Investigations of Collaborative Design Environments

A Framework for Real-time Collaborative 3D CAD

A thesis submitted for the degree of Doctor of Philosophy

by

Tek-Jin Nam

Department of Design, Brunel University

February 2001

To my family

Abstract

This research investigates computer-based collaborative design environments, in particular issues of real-time collaborative 3D CAD. The thesis first presents a broad perspective of collaborative design environments with a preliminary case study of team design activities in a conventional and a computer mediated setting. This study identifies the impact and the feasibility of computer support for collaborative design and suggests four kinds of essential technologies for a successful collaborative design environment: information-sharing systems, synchronous and asynchronous co-working tools, project management systems, and communication systems.

A new conceptual framework for a real-time collaborative 3D design tool, Shared Stage, is proposed based upon the preliminary study. The Shared Stage is defined as a shared 3D design workspace aiming to smoothly incorporate shared 3D workspaces into existing individual 3D workspaces. The addition of a Shared Stage allows collaborating designers to interact in real-time and to have a dynamic and interactive exchange of intermediate 3D design data. The acceptability of collaborative features is maximised by maintaining consistency of the user interface between 3D CAD systems.

The framework is subsequently implemented as a software prototype using a new software development environment, customised by integrating related real-time and 3D graphic software development tools. Two main components of the Shared Stage module in the prototype, the Synchronised Stage View (SSV) and the Data Structure Diagram (DSD), provide essential collaborative features for real-time collaborative 3D CAD. These features include synchronised shared 3D representation, dynamic data exchange and awareness support in 3D workspaces. The software prototype is subsequently evaluated to examine the usefulness and usability. A range of quantitative and qualitative methods is used to evaluate the impact of the Shared Stage. The results, including the analysis of collaborative interactions and user perception, illustrate that the Shared Stage is a feasible and valuable addition for real-time collaborative 3D CAD.

This research identifies the issues to be addressed for collaborative design environments and also provides a new framework and development strategy of a novel real-time collaborative 3D CAD system. The framework is successfully demonstrated through prototype implementation and an analytical usability evaluation.

Acknowledgements

I would like to express my sincere thanks and appreciation to my supervisor, Dr. David Wright, who has provided direction, assistance and encouragement during my study. I also appreciate the opportunities he has provided me to continue learning and teaching in the UK. I would like to thank Professor Peter Isherwood and the Design Department at Brunel University for providing me an excellent research environment. I gratefully acknowledge the financial support from the Department and from the UK government through the Overseas Research Studentship Awards.

I wish to express my gratitude to the all members of the Interaction Design Research Group where I spent most of my time during my postgraduate study, for providing me with delightful learning and working atmosphere. In particular, I would like to thank Dr. Seungfeung Qin, Matthew Benham and Paul Gammans for their caring fellowship. I am very grateful to Brunel design students who voluntarily participated in my experiments.

I would like to thank professionals from the CSCW and Design Research community, Stephen Scrivener, Chengzhi Peng, Mark Perry, Atul Prakash, and Joerge Haake, for the valuable feedback and the constructive advice about the ideas presented in this thesis. Thanks to my colleges at the School of Product and Engineering Design, University of Wales Institute Cardiff, for their assistance. I am very grateful to Dr. Ivan Jordanov and Steve Gill for their assistance during my preparation for publication.

Special thanks to Dr. Kyungwon Chung, the president of Korea Institute of Industrial Design Promotion, for his continual assistance and encouragement since my undergraduate study. I would like to thank our parents for their encouragement, support and understanding. Finally, I would like to express my sincere gratitude to my wife, Hyunju Lee, for her love, encouragement and sacrifice.

Table of Contents

Chapter 5. A Real-time Collaborative 3D CAD System: Syco3D 97

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Team approaches in design

Design activities involve team effort. As the complexity of design problems has increased, it has become impossible for a single designer to manage all aspects of a design project. Design activities have become collaborative and multi-disciplinary team activities. Pugh explained the importance of a team approach in design as follows:

> Design is a team activity requiring the creative integration of specialists. Diversity within a team is crucial, but diversity needs to be carefully handled. What is needed above all else in the context of design is the use of systematic methods which provide a structure so that disagreements converge productively onto solutions all can understand and all can accept. (Pugh, 1996:p336)

There is a growing interest in tools to support collaborating professionals working in teams. There is particular emphasis on tools and methods to support distributed team work in the business environment. As geographical and cultural distances appear to shrink with the advent of technical advances in transport and telecommunication, companies are trying to take advantage of international marketing opportunities. Increasingly, the resources of companies are being distributed widely. For example, the design and production departments of a company may be located in several countries. It is common for designers to collaborate in teams across physical, departmental and even company borders. Tools and methods to co-ordinate resources effectively are critical in this situation.

It is essential to support collaborating designers efficiently. To date tools used in the design domain have not provided sufficient support for team based design activities. Computer based design tools have helped designers in many ways, such as visualising the design concept, analysing design problems and simulating design solutions. However, most of the computer based tools currently available have been developed specifically for supporting single user environments.

To address this situation, in this thesis tools and methods are investigated to provide better support in collaborative design environments. In particular, it is focused on one kind of collaborative design tool, a real-time collaborative 3D CAD system. The investigation applies the knowledge of Human-Computer Interface (HCI) and Computer Supported Co-operative Work (CSCW) into the design domain, from the designer's perspective.

The remainder of this chapter will provide some background on the computer based tools used in the design domain, introduce issues in CSCW and HCI and their relation to the design domain, state the research problems and goals of the research and show the organisation of the thesis.

1.2 Computer based tools in the design process

Computer based design tools have offered a new dimension in the design process. Due to their introduction to the design process, designers now have a wide variety of media at their disposal, such as digital images, hypertext and multimedia as well as traditional pen and paper based media. In addition, computer based tools are used in many phases of the design process in supporting design activities. 2D graphics and CAD tools are used in the early phases of the design process, where designers want to focus on visualising design concepts. A set of graphic design software originally developed to computerise the 'design for print' process became popular with product designers. These tools include illustration, desktop publishing and image manipulation applications and are used at various stages in the design process, for such tasks as rendering, image editing and designing and specifying product graphics. Designers also use multi-media authoring tools to simulate and test the interfaces of electronic products. Design database and information browsing systems are used throughout the design process, for instance where designers are required to investigate production processes, materials, or even the marketing of a product.

There is no doubt that 3D CAD systems play an important role in design activities, because many design solutions are realised in the form of a 3D artefact. In product design, 3D CAD systems help designers by computerising the process from early concept generation to detail development and manufacturing. These tools allow experimentation with such features as angle of view, colour, surface finish, lighting, product graphics and various structural properties without fear of losing the original concept. 3D design concepts can be represented by wireframe, surface and solid 3D models. Tangible prototypes can be produced automatically using rapid prototyping tools, such as 3D printing and stereo lithographic systems. Effectively implemented 3D design tools are not only for product designers but also for other professionals participating in general product development. In this respect, 3D CAD tools are one of the most important computer based tools for product designers.

One of the problems with existing computer based design tools, particularly more complex tools such as 3D CAD systems, is that designers have difficulties using these tools efficiently. The difficulties may be caused by the complicated and unnatural user interfaces of existing 3D design tools. Developers of design tools sometimes build the tools without having a full understanding of how designers work. As a result, designers have to change their work patterns in order to match the interface that the tools may require, although this may not be the most efficient and natural way to accomplish a given design task. Therefore users' perspective on the development of design tools is essential.

1.3 Computer Supported Cooperative Work and Design

Most of the computer based tools mentioned above have been developed largely to support single user environments. The rapid development in network and computer technologies provides new opportunities to transform these single user oriented design tools to multi-user equivalents. This new generation of computer based design tools can be developed by applying the techniques and theories of Computer-Supported Co-operative Work (CSCW) into the design domain. CSCW is defined as computer assisted co-ordinated activities carried out by a group of collaborating individuals (Wilson, 1991; Baecker, 1993; Grudin, 1994). The information technology of CSCW used to help people work together more effectively is called Groupware (Johansen, 1988).

The scope of CSCW is very broad because most human activities involve some sort of collaboration and team work. This is reflected in the wide range of systems that have been designed, built, and studied. For example, these have included; shared whiteboard/meeting room systems such as Dolphin (Streitz et al. , 1994) and Colab (Stefik et al. , 1987a), workflow/process-based systems such as The Coordinator (Winograd, 1987), authoring systems such as Quilt (Leland et al. , 1988) and SASSE (Baecker et al. , 1994), drawing systems such as Commune (Bly and Minneman, 1990; Minneman and Bly, 1991) and The Conversation Board (Brink and Gomez, 1992), collaborative virtual reality systems such as MASSIVE (Greenhalgh and Benford, 1995), large scale database systems and others. CSCW is also concerned with research into behavioural foundations of group activity (Suchman, 1987), group interaction in natural settings such as air traffic control rooms (Harper and Hughes, 1993), as well as asynchronous (Sproull and Kiesler, 1991) and synchronous communication (Egroff, 1991).

CSCW is considered to be a sub-area of Human-Computer Interaction (HCI). It has drawn from a wide range of multi-disciplinary perspectives on its own subject matter. While HCI has focused on the interface between individual users and computer systems, CSCW and Groupware have focused on human-to-human interfaces. As Human

Computer Interaction (HCI) uses cognitive and experimental psychology in addition to computer science, interface design and software engineering, much of the development of CSCW has been influenced by research in anthropology and sociology.

The technologies of CSCW and groupware systems provide a new collaborative environment in the design domain. For example, video conferencing systems currently available allow distributed designers to accomplish a design project without actually meeting face-to-face. Concurrent engineering tools and methods help a product development team work together more effectively. Using database sharing and file transferring tools, designers can exchange various information more rapidly and accurately. Designers can collaboratively review complex 3D models or multi-media presentations from remote sites. Physical constraints are becoming less critical for collaboration between designers.

1.4 Research problems

Two research problems are identified in the application of CSCW techniques and theories in the development of a new generation of collaborative design tools: i) a limited understanding of collaborative design environments and the impact of new tools, and ii) a lack of appropriate tools to build a better collaborative design environment, in particular to support real-time collaborative 3D CAD.

Firstly, the development of collaborative design tools should be based on a sound understanding of team design activities. Studying how designers work together in various situations informs us of the new opportunities and challenges offered by collaborative design tools and environments. Investigation into the impact of collaborative technologies in team design activities also provides a valuable insight for the next generation of collaborative design tools. Although there have been attempts to study team design activities in various situations, because of the complexity and variety of design activities, our understanding of them is incomplete. Investigations should be carried out to identify how collaborative technologies can be used to support collaborating designers and their impact on team design activities.

Secondly, there is a lack of effective tools for a collaborative design environment. Some computer-mediated tools and database sharing systems are in use by designers, but such systems were originally designed for general purpose group activities. In particular, among the tools used to support collaborating designers computer aided real-time collaborative 3D design tools have not drawn much attention. Complexity and a lack of supporting methodologies limit the development of such tools. Most 3D design tools have assumed that only one person would accomplish a required task. However, there are many situations where concurrent co-operative designing with 3D tools can be effective (Jasnoch et al., 1994; Greco, 2000; Hewlett Packard Company, 2000). For example, when a design model is complex and consists of many parts, a person in charge of the 3D CAD task has to make a sustained effort just to construct an initial 3D CAD model. As the model is built by only one person in a team, the following process is likely to involve multiple modifications of the initial model. Once the CAD task is accomplished by one person, the subsequent modifications are difficult to share. Each modification may require substantial time and effort for the person who originally built the model and this could delay the whole design process. In such a situation, if designers or professionals involved in the product development process could work together synchronously to discuss and modify the evolving 3D CAD output, the efficiency of the team design process would be greatly improved (Gisi and Sacchi, 1994).

These research problems are interconnected. In order to develop an appropriate tool for computer aided real-time collaborative 3D design, it is essential to have a good understanding of the 3D design activities as well as the overall design process. The computer aided real-time collaborative 3D design tool is one component of a collaborative design environment. Therefore, ways of integrating collaborative design tools should be considered in order to maximise the usefulness of the new generation of design tools. The research problems are also related to three main research areas; CSCW, design research and the development of computer applications. Figure 1.1 shows how the related research fields overlap. This thesis addresses these research problems.

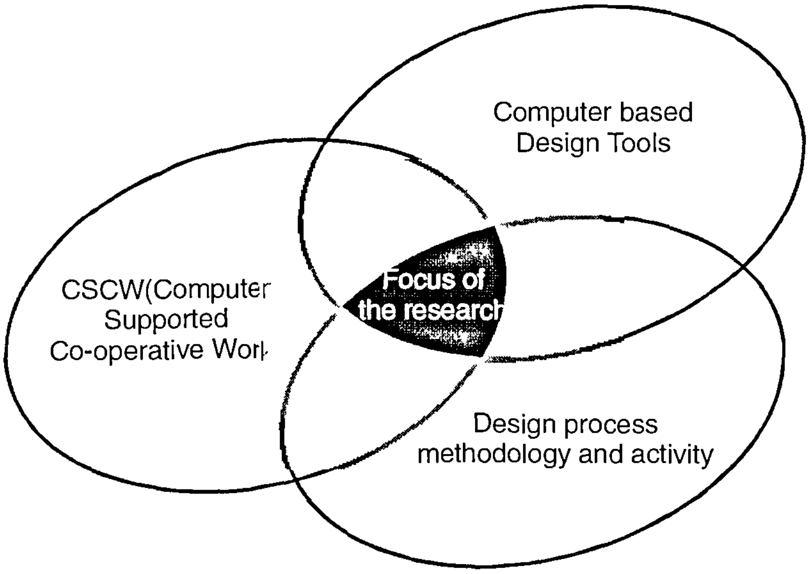


Figure 1.1 The research fields of collaborative des'gn rese' rch

1.5 Research objectives and methodology

The research aims to improve our understanding of the collaborative design environment, and to investigate one kind of collaborative design tool, a computer aided real-time collaborative 3D design system. Three main objectives include:

1. An improved understanding of collaborative design activities and investigation of the impact of collaborative technologies in the collaborative design process.
2. An investigation of an operational framework and a prototype system from a user's perspective, focusing on real-time collaborative 3D CAD.
3. An evaluation and analysis of the feasibility and the impact of the proposed system in a real-time collaborative design environment.

The first objective of the research is an improved understanding the ways in which designers work together in a traditional setting and to investigate the impact of current collaborative technologies in team design processes. This objective is met by a preliminary study of team design projects under two different experimental settings: one with a series of face to face meetings and the other with computer mediated meetings. From the preliminary study, an understanding of the resources and problems of collaborative design environments is gained.

Next, the research focuses on one of the most unique design activities, which involves computer aided 3D design tasks. The research aims to provide an operational framework for a new collaborative design tool for this task. Particular interest is put on the shared 3D design workspace and ways to provide smooth incorporation of an individual and shared 3D design workspace. It is based on lessons learned from the preliminary study and my own experience of using CAD systems as a designer. Collaborative conceptual 3D design tools are selected because 3D modelling is one of the most important activities in the design process for industrial designers. The research examines issues related to the real-time collaborative 3D design activities and demonstrates the feasibility of the framework by implementing a series of research prototypes.

The final objective of the research is to evaluate and analyse the impact of the new real-time collaborative design tool. This is met by a usability experiment that examines the usability and the usefulness of the framework and the resulting research prototype. It also identifies further issues to be considered for the development of future generations of collaborative design tools.

In order to achieve these objectives, the research takes an evolutionary approach, which involves an observational study, investigating and proposing a conceptual model for a new tool, implementation of a prototype system based on the proposed conceptual model and evaluation. Figure 1.2 shows the steps taken to achieve these three research objectives and the scope of the research. The first step of the research is to understand collaborative design activities. Then the framework and the research prototype is developed. Designers' activities may be changed through the introduction of a new prototype and the impact is examined in the last phase of the research. Insights on a new generation of collaborative design tools are gained by integrating the findings of all steps.

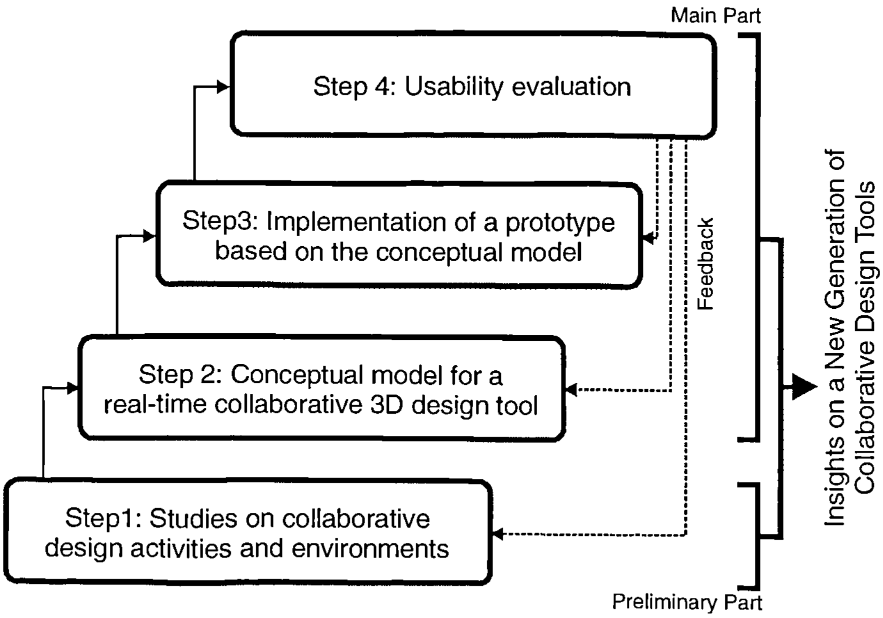


Figure 1.2 Four steps to achieve the research objectives

1.6 Organisation of the thesis

The remaining part of the thesis is organised in the following way: **The second chapter** reviews work related to the two main research problems mentioned above. The related studies have been categorised into four main areas. The first area relates to an understanding of team design activities for developing tools. The studies of group activities provide an insight into the tools supporting the activities. In the second related area, a selection of research on general 2D shared workspace tools is reviewed. Many issues raised in the shared 2D workspace research can be extended to a 3D collaborative environment. The third area is concerned with shared 3D workspace tools. Investigations of previous computer-based collaborative 3D tools are reviewed. Finally, a selection of work on 3D interaction techniques is reviewed to consider user interface techniques for a shared 3D design workspace.

The third chapter reports on the preliminary study of a team design project. The main goal of the study is to understand the process of team design projects and the ways in which computer-based design tools and collaborative technologies can be used in a collaborative setting. Experimental design projects, accomplished by groups of designers in two different situations, a conventional face-to-face team design environment and a computer mediated design environment, are observed and analysed.

The fourth chapter considers the conceptual and technical basis for the development of a real-time collaborative 3D design tool. The concept of shared 3D workspace in the design domain is examined by comparing it with a general shared workspace. Current 3D design applications are analysed in terms of the process using them and the limitations in a team design project. The requirements and goals of the new collaborative 3D CAD system are specified based on the analysis. Then a framework of Shared Stage is suggested. A smooth integration between individual and shared workspace has been emphasised in the framework. The technical basis of the development of collaborative application is introduced and development environments and tools that are directly related to the prototype implementation are also discussed.

Chapter 5 describes the evolution of our research prototype in three main steps. The initial prototype is based on a commercial 3D CAD system and developed as plug-in application to be loaded into a host system. The next prototype separates the collaborative features from the host CAD system. The final prototype is developed as a new multi-user 3D CAD system that dynamically incorporates the shared 3D workspace. The essential features of a 3D CAD system and real-time collaborative features provided through the Shared Stage are illustrated as well as the details of implementation.

Chapter 6 presents a usability evaluation using the research prototype. The experiment investigates the way that design teams use the system in semi-realistic collaborative 3D design activities. It compares the collaboration-aware version of the prototype with a conventional single-user version. Quantitative and qualitative analysis are presented to examine the feasibility of the Shared Stage and lessons learnt from the evaluation are discussed.

Chapter 7 compares the result of this investigation with different methods reviewed in Chapter 2 and discusses further issues raised from the implementation and evaluation of the new real-time collaborative 3D CAD system. In particular, issues are discussed in terms of the relation of the real-time collaborative 3D design tool to collaborative design environments and 3D interaction techniques in a shared 3D workspace. The application of the World Wide Web and VRML as a platform for the real-time collaborative 3D system is considered as well as 3D workspace awareness issues.

In **the final chapter**, the research questions and the methods that are addressed in this research are reconsidered. Major contributions are illustrated with a brief summary of the findings for each contribution.

Chapter 2

Review of Related Works

2.1 Introduction

In this chapter, I review earlier work on tools and environments for collaborating designers. I begin by reviewing studies with a broad perspective on collaborative design activities and environments. I then focus on works related to the focus of this thesis, real-time collaborative design tools to support shared design workspace activities. The review is classified into four sub areas: research into team design activities, shared 2D workspace, shared 3D workspace and interaction techniques for 3D workspaces.

The first area is concerned with a social and psychological foundation for the development of tools used in a collaborative design environment. Researches about team design activities and computer support for collaborative design are reviewed to improve understanding of the design process and the activities. The second section presents a review of shared 2D workspaces. The tools for shared 2D workspaces are often designed to support general group activities. It is essential to understand these general issues to develop collaborative design workspace tools. Considerable research has been carried out regarding general group tools in shared 2D workspaces in the field of CSCW. The review focuses on how the findings of previous studies can be applied to the shared 3D design environment. In the following section, studies that focused on collaborative 3D visualisation and multi-user CAD are reviewed in the area of shared 3D workspaces. Some research in this section has similar objectives to our work since it aimed to provide a new multi-user 3D tool. Comparisons of the approaches and the findings are presented in Chapter 7. Finally, 3D interaction techniques used in computer-based design tools are reviewed because natural interface with collaborative 3D design tools is important to improve the usability of the new tools. Related works are reviewed for new 3D interaction techniques that can be extended to collaborative 3D design workspaces.

2.2 Studies on design activities

Earlier empirical studies were primarily concerned with studying design processes. Some researchers attempted to systematically analyse design activities, as it is one of the highest cognitive activities of human beings (Simon, 1981). Some were interested in design processes and methods in order to prescribe a systematic design process (Asimow, 1962; Jones, 1970; Alexandar, 1964). Many of these early studies were criticised for being too simplistic and divorced from actual design practice. Subsequent work sought to understand and describe more of the complexity of design activities. Recently there has been a growing number of investigations examining design processes and activities in order to build tools to be used by the designers involved. In particular, some investigations were concerned with collaborative design activities and tools to support the activities. A selection of these studies is presented here.

Two Delft workshops have suggested ways to study design activities and produced some interesting results (Cross et al., 1992; Cross et al., 1996). The latter workshop brought together a distinguished group of design researchers to compare analysis methods of the same data, and to discuss the state of the art in protocol analysis. The data prepared for analysis at the workshop were the recordings of an individual designer and a three-person team of designers working for two hours designing a typical industrial design project. The task was to design a 'fastening device' that enabled a given backpack to be fastened onto a mountain bike. Each researcher employed their own research methods to understand the design activity. Some researchers employed protocol analysis and the think aloud method (Akin and Lin, 1996), while others used observational and descriptive examination on the design activities (Cross and Cross, 1996). Some researchers focused on designers discourse (Goldschmidt, 1996), while others focused on drawing activities (Mazijoglou et al., 1996), or the management of design information (Baya and Leifer, 1996). As a result of the workshop, an overview of the accumulated knowledge on design of these researchers was produced and the protocol analysis was validated as a research technique for design to some extent.

One of the researches in the Delft protocol workshop looked at the role of 3D objects in

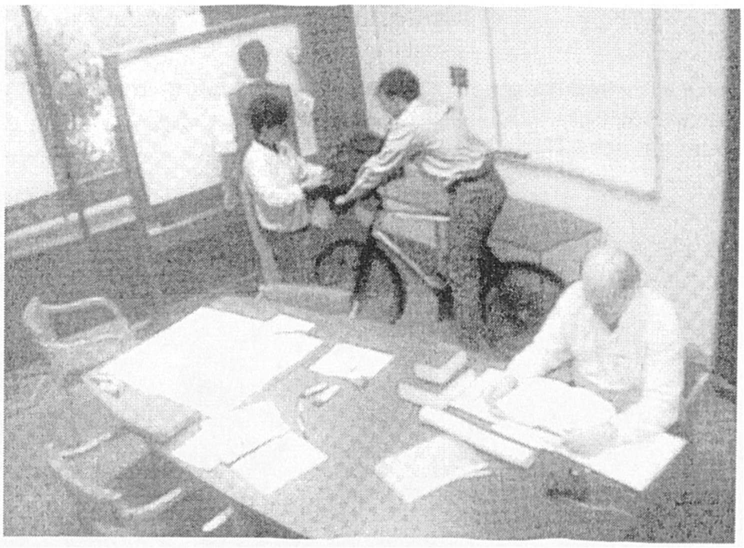


Figure 2.1 An experiment video of the Delft Protocol Workshop (Harrison and Minneman, 1996: p431)

a design workspace. Harrison and Minneman (1996) investigated how 3D objects are used in the experimental design project. They intended to apply their findings to design systems that adequately support the full range of interactions that occur around and with 3D objects. They adopted a method of continual refinement (Minneman, 1991; Strauss, 1987) for the analysis of interaction that engaged with 3D objects. They suggested that 3D objects are more than a source of information being constituents of the activity and frames for communications. They pointed out that 3D objects alter the dynamics of interaction, especially in multi-designer settings. It suggested that 3D objects and interactions with and around 3D objects play an important role in collaboration, not just being an outcome of the activity.

Social science researchers have also conducted field studies to investigate design meetings. Olson et al. (1992) presented details of how real design groups engage in early software design meetings. They were interested in the design of computer systems to support co-operative work in a meeting. From ten design meetings of four projects in two organisations, they developed a coding scheme to analyse participants problem-solving activities and the structure of their design arguments. They found similarities in how people spent their time and in the sequential organisation of the activity. As one of the implications for supporting tools, they pointed out that aids for structuring the activity might not be necessary, because design discussions were already structured. They also suggested that the design categories used in their analysis might map onto the analysis of other kinds of problem-solving meetings.

One of the most influential studies of tools in the shared drawing workspace was conducted by Tang (Tang and Leifer, 1988; Tang, 1989; Tang, 1991). He investigated a shared workspace activity of small groups working on conceptual design tasks by examining eight sessions of short, small group, conceptual design activity. The experiment task was to design a remote controller. He presented a methodology for observing and analysing design activity based on empirical methods used in ethnographic and interaction analysis. As a result, he identified two dimensions of workspace activity: actions and functions (Figure 2.2). The actions describe the process of producing the activity: listing, drawing or gesturing. The functions indicate what purpose the activity effectively accomplishes: storing information, expressing ideas, or mediating interactions. From the descriptive analysis of workspace activity, he proposed six design implications for collaborative technology. His specific recommendations for the design of tools to support shared workspace activity include:

- Providing ways of conveying and supporting gestural commination;
- Minimising the overhead encountered in storing information;
- Conveying the process of creating artefacts to express ideas;
- Allowing intermixing of workspace actions and functions;

- Enabling all participants to share a common view of the workspace while providing simultaneous access and a sense of close proximity to it;
- Facilitating the participants' ability to co-ordinate their collaboration.

Action / Function	LIST	DRAW	GESTURE
Store information			
Express ideas			
Mediate interaction			

Action

- List – actions producing non-spatially located text; alpha-numeric notes.
- Draw- actions producing graphic marks and objects, including spatially located textual annotations.
- Gesture- purposeful body movements which communicate information, such as referring to existing objects in workspace or enacting simulations.

Function

- Store information – preserving information in some form for later recall, typically after attaining explicit group agreement.
- Express ideas – interactively creating representations of ideas in some tangible form, to enable the group to perceive, react to, and build on them.
- Mediate interaction – facilitating the collaboration of the group, such as moderating the turn-taking or directing the group's attention.

Figure 2.2 Framework for analysing workspace activity (Tang, 1989:p67)

In addition to investigations of design activities, the impact of communication technology has been perceived by many researchers, and computer support for collaborative design has grown into a major area of design research (Maher et al., 1999; Maher et al., 1997; Engeli and Mueller, 1999; Saad, 1994; Saad and Maher, 1996 ; Schmitt et al., 1997; Haymaker et al., 2000). Early studies focused on ways of managing and sharing design information among a number of professionals involved in a team design project. The focus is now broadened to ways of improving interaction between collaborating designers. For example, Toye et al., (1993) explored a collaborative product development environment for product development teams by incorporating heterogeneous software applications. A number of researchers (Maher et al., 1999; Yee et al., 1998; Wojitowicz, 1995) examined the concept of a distributed collaborative design environment in a series of Virtual Design Studio projects, in which (student) designers around the world collaborate with each other through communication and sharing of design ideas using various software tools. Various technical and social issues for computer supported collaborative design environments have been raised by the idea of a virtual design studio (Maher et al., 1999). There are technological difficulties in establishing a suitable environment for sharing information and these issues become greater when collaboration at a distance is the only possible means of exchanging and communicating information. Social problems arise from a lack of understanding of the differences in working remotely and working face-to-face. Both technical and social issues need to be fully addressed to maximise the usefulness of the new technologies for design activities.

With regard to the interaction between collaborating designers, Maher et al. (1998) compared an individual and a collaborative design setting looking at the amount and content of design semantics which were documented using computer based communication tools. Three teams of two designers participated in the experiment and were asked to solve an architectural design problem alone in the first session and to design collaboratively in the second session (Figure 2.3). They used video conferencing software, a shared whiteboard, and the Timbuktu multi-user environment for application sharing. As a result, they found that designers tended to document more information related to the purpose of their design during the non-collaborative sessions than during collaborative work. They also found that a valuable amount of the semantic information is left undocumented due to the intensive exchange via video conferencing. They observed that there were three different design styles: close-related work, independent work, and work dictated by a leader.

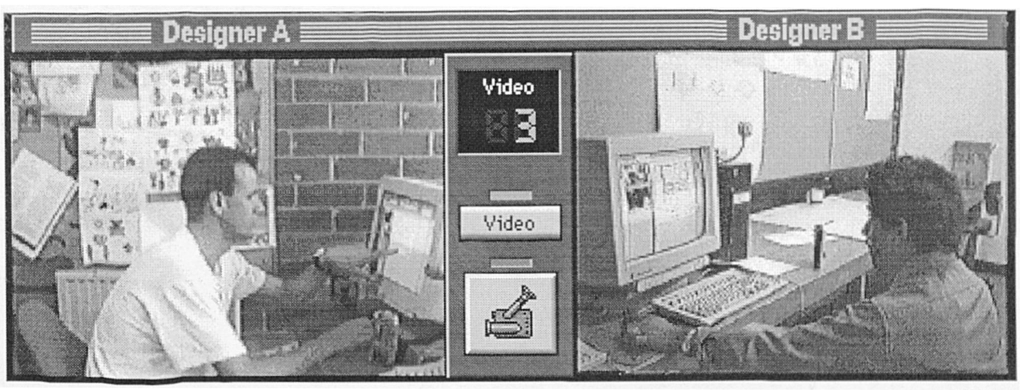


Figure 2.3 Experiment of computer mediated design (Maher et al., 1998)

2.3. Studies on shared 2D workspaces

In the area of CSCW, a number of real-time collaborative applications have been studied to address various technical and human interface issues in shared 2D workspaces for general meetings. These applications can be classified into three types: collaborative writing, shared sketching and object-oriented 2D drawing. Many researchers have studied collaborative writing to develop an example system of distributed collaborative applications. Some collaborative writing applications provide asynchronous support for groups that work in a sequential manner, passing versions of the document around between group members, e.g. Quilt (Leland at al., 1988), PREP (Neuwirth at al., 1994). Other systems support a mix of synchronous and asynchronous collaborative document editing, e.g. CES (Grief et al., 1992), Duplex (Pacull at al., 1994). There have also been quite a few systems developed to support synchronous collaborative writing, e.g. GROVE (Ellis et al., 1991), ShrEdit (McGuffin and Olson, 1992), and SASSE (Baeker et al., 1993; Baecker et al., 1994).

A number of tools have been developed to support collaborative drawing in shared 2D workspaces. Some to support simple shared sketching, while others were developed to support more specific 2D tasks, such as diagramming or producing conceptual maps. Although these applications deal with shared 2D workspaces, their interfaces can be quite different from one another. While a simple shared sketching task is considered as a whiteboard in a general meeting, object-oriented shared drawing systems can be considered as a drawing table of designers. The activity and interface at the whiteboard is simple drawing and text writing without significant or detailed modification of the graphics. Whereas the interface of the object-oriented shared system requires a series of functions to accomplish a sophisticated drawing task. A selection of collaborative drawing tools in two categories are reviewed as some construction principles can be applied to design tools in shared 3D workspaces.

Tivoli (Moran et al., 1995) is one of the object oriented shared 2D workspace tools designed by Xerox PARC to support informal workgroup meetings. It allows collaboration where the participants are co-present or remote. To achieve independence, unique identifiability and immutability, Tivoli was developed as an object graphics model. The following ten design principles were employed for the design and implementation of the sharing functionality in Tivoli.

- Object-level principles
  1. Maintain object independence and the additivity of operations
  2. Use the immutability of objects to prevent inconsistencies
  3. Alert users to conflicting concurrent operations so that they can resolve them
- Structure-level principles
  4. User IDs to prevent ambiguous references
  5. Use sufficient structural descriptions of operations to detect potential inconsistencies
  6. Resolve structural inconsistencies by giving sites a priority order
- Interaction-level principles
  7. Broadcast gestures to prepare remote users for actions
  8. Keep remote sites updated at frequent intervals on the partial creation of objects
  9. Keep remote sites updated at frequent intervals on where the user is pointing
- Independent work principles
  10. Provide private workspaces where individual users can work by themselves

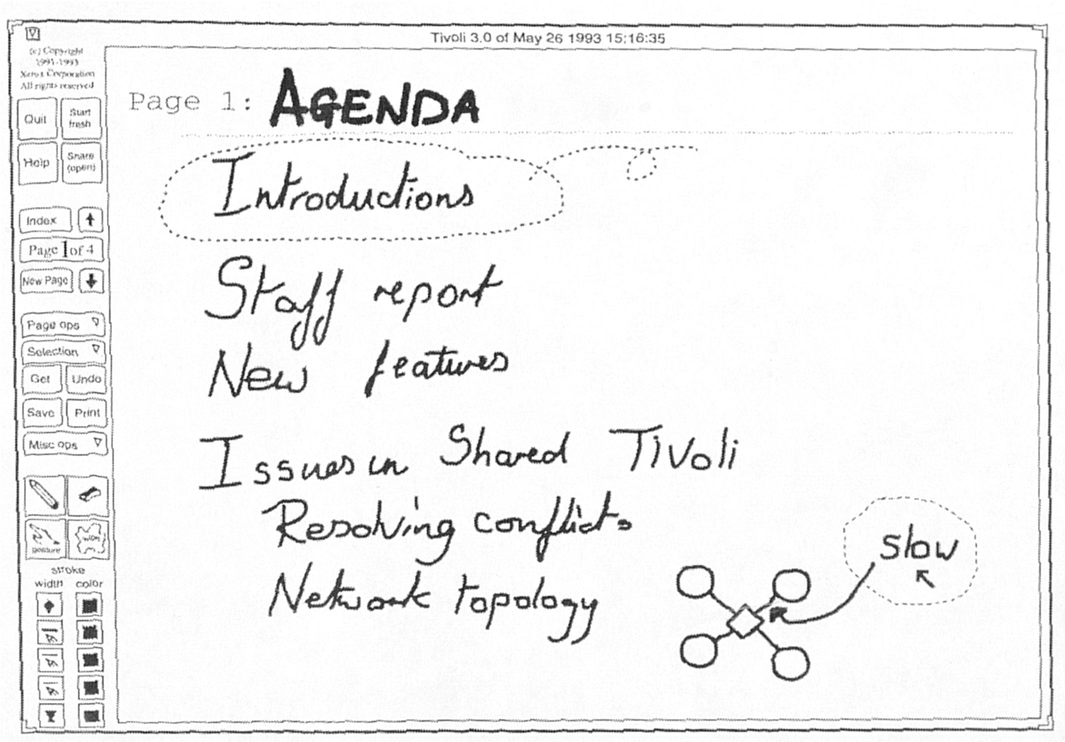


Figure 2.4 Screenshot of Tivoli display(Moran et al., 1995)

Roseman and Greenberg (1996) reported their experiences of building systems to support remote real-time group interaction: GroupSketch and XGroupSketch, both multi-user sketchpads; GroupDraw, a prototype object-based multi-user drawing package; and GroupKit, a groupware toolkit. They used Tang's six criteria (Tang, 1991) as a foundation for the system. GroupSketch developed as a simple group sketching tool, and its main features included WYSIWIS (What You See Is What I See) display, multiple active cursors, simultaneous interaction, and modeless support of gesturing and listing. While GroupSketch and XGroupSketch are paint programs where users can only make and erase marks on a bitmap surface, GroupDraw is an object-oriented drawing program similar to structured drawing packages, such as MacDraw. Their experience with GroupDraw raised several specific interface issues for object oriented shared drawing systems. The first issue was the access control problem occurring when several people try to manipulate an object. In GroupDraw, if an acquisition conflict does occur and permission is denied, the object will snap back to its original status. The second issue they raised was the seamless intermixing of workspace actions and functions. This issue was not fully addressed bacause GroupDraw users must select from a variety of object types and go into a particular drawing mode. The third issue was the need for a private work surface. A scrollable drawing surface and attaching coupling status to objects was suggested as a strategy to address this problem. Finally, they raised the problem of scrollable drawing surface that disrupted the WYSYIS (What You See Is What I See) principle. As a strategy to mitigate this problem, they suggested a separate radar window showing a miniaturised animated view of the complete drawing surface and the location of all participants' viewports on it. View-slaving, where the viewport of one participant can be enslaved to the viewport of another participant was also suggested. They also pointed out the need for a better sense of tele-presence, the ability to have a seamless modelling of shared work on the computer and the desktop, and consideration of the size of the group.

Trade-offs in the choice between replicated, centralized and hybrid architecture were pointed out and it was suggested that the choice of a style would often depend upon the physical requirements of the system. They recommended that conference registration be managed independently from the underlying application. They considered multiple cursors to be fundamental to these systems.

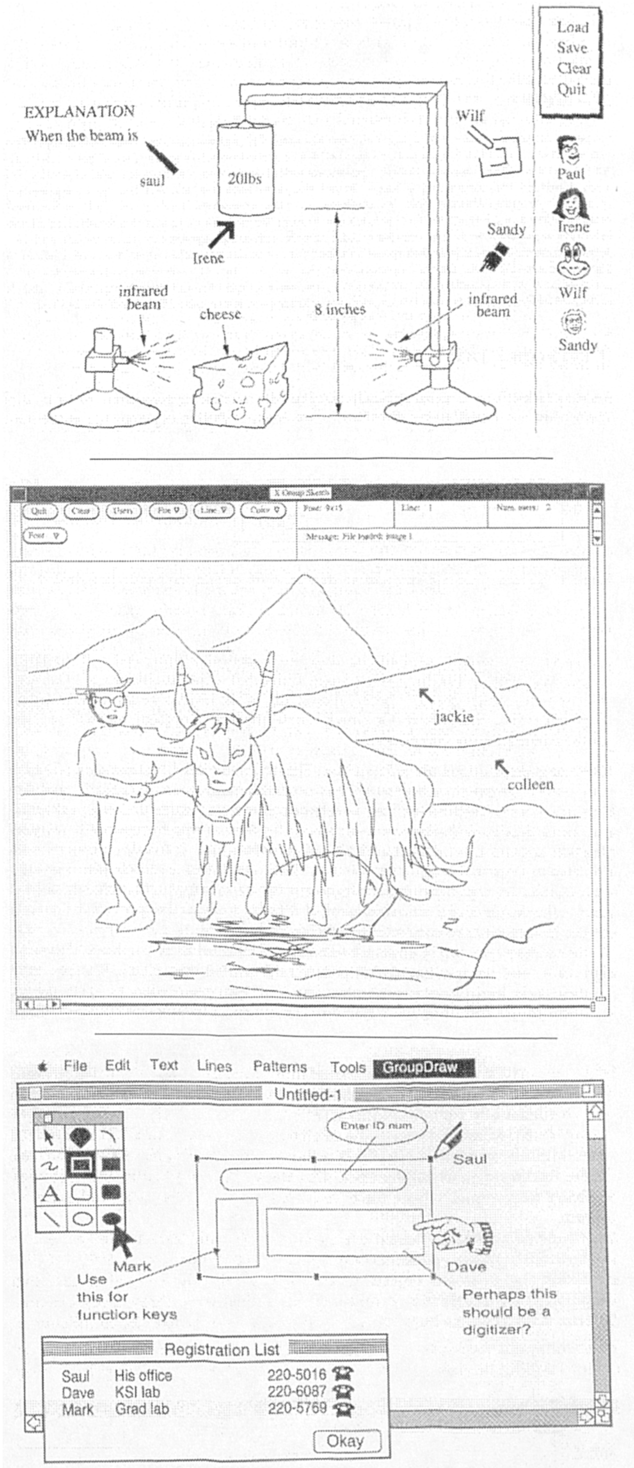


Figure 2.5 Screenhot of Groupsketch(Top), Xgroupsketch(Middle) and GroupDraw(bottom). (Roseman and Greenburg, 1996)

Tang and Minneman reported the design, implementation and the use of a shared drawing tool using video (Tang and Minneman, 1991). Based upon their observations on shared drawing activity, they identified three aspects that have implications for tools to support the activity: i) hand gestures are used prominently and productively, ii) timing relationships help the participants understand the drawings created, and iii) timing relations and spatial arrangement help the participants negotiate their use of the shared drawing surface. These were incorporated into their prototype, which conveyed hand gestures, did not disrupt timing relationships, offered a new sense of spatial relationships and allowed concurrent access to the shared space. Several limitations were also reported with the use of the prototype. These included the limitation of the size of the workspace to be shared, the lack of access to a partner's drawing, and parallax and clarity difference between the drawings on the screen surface and on the video. The VideoDraw concept (Figure 2.6) provided the basis for a few other similar studies using a video as the medium of shared 2D workspace.

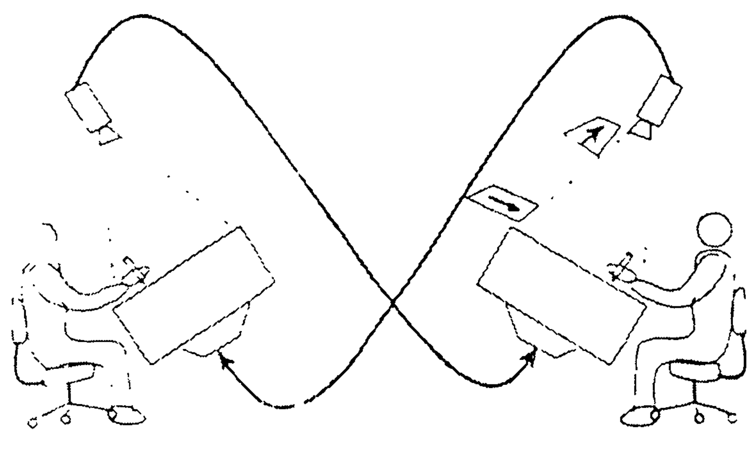


Figure 2.6 Schematic diagram of VideoDraw connecting remote locations (Tang and Minneman, 1991)

Ishii et al. (1995) presented the evolution of the novel shared drawing medium ClearBoard, which allows co-workers in two locations to draw with coloured markers or with electronic pens and software tools while maintaining direct eye contact and employing natural gestures. Their research progressed through iterative design steps from TeamWorkstation-1 (Ishii, 1990) and TeamWorkstation-2 (Ishii et al., 1993), to ClearBoard-1 (Ishii, 1992) and ClearBoard-2 (Ishii et al., 1995). Early TeamWorkstation prototypes provided the video image of participants faces and the shared workspace but suffered form an undesirable seam between the face images and the shared workspace. In order to provide dynamic and interactive focus switching between shared workspace and interpersonal space, they employed a metaphor of a transparent glass board. Based upon the experience of the designs and experiments of the ClearBoard prototypes, they raised several issues, including multi-user and multi-point support of these prototypes, new display technology that make the ClearBoard concept accepted, and interpersonal distance. They emphasised the importance of gaze awareness, which includes eye contact and monitoring the partner's direction of gaze. They also suggested that ClearBoard-2 integrated the technology of computer-based groupware with that of video conferencing.

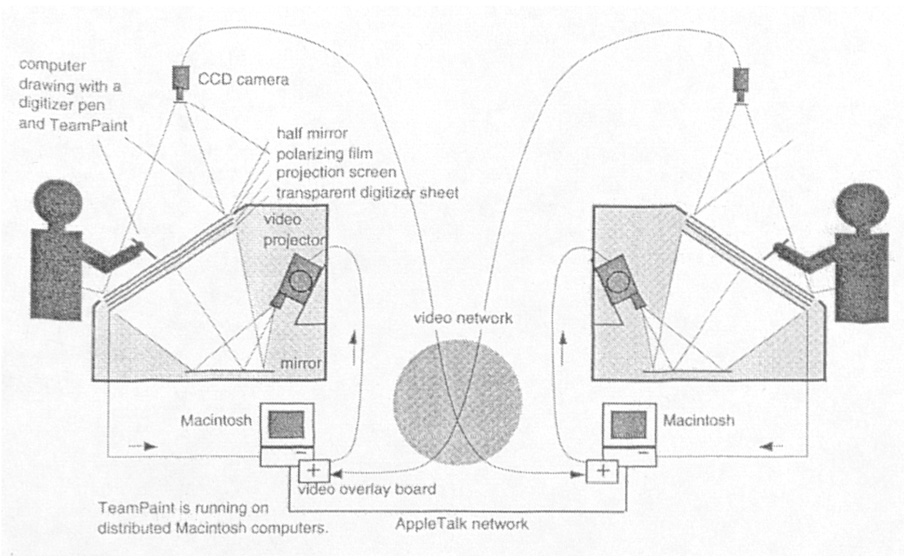


Figure 2.7 System Architecture of ClearBoard-2 (Ishii et. al., 1995)

Scrivener et al. (1993) reported an experiment in which teams of two designers used a shared drawing tool, called a ROCOCO sketchpad, for conceptual design. They examined the capability of the sketchpad and the problems encountered when parties are geographically separated, between the UK and Australia. They concluded that the sketchpad was usable and useful for early meetings of design work at a distance. They suggested the need of the democracy of control and interaction fluency to support synchronous remote design meetings. They also raised the issue of group communication about views of objects generated with single user applications.

2.4 Work related to shared 3D workspaces

Comparatively little consideration has been given to the shared 3D workspace in the area of CSCW. Investigation of 3D virtual space is important for several reasons in the area of design and CSCW. Researchers consider the 3D environment as a medium of collaboration. They have tried to address social and communication issues by 3D elements in the environment, and Collaborative Virtual Environment (CVE) is one result. In CVE, for example, proximity can be easily displayed in 3D space. However, there are many issues to be addressed since the user interface is sometimes very limited and interaction within CVE with current interaction devices is unnatural (ActiveWorlds, 2000; Stenius, 1996). Secondly, 3D virtual space is important for designers because collaborative activities often take place with or around 3D workspace. There are many situations where 3D models become a main outcome, such as conceptual and engineering 3D design and scientific visualisation. Because users have to deal with 3D in their task, collaborative interaction with 3D objects or workspace is essential. This section presents a selection of these studies from several areas, ranging from engineering and architectural design to computer graphics and visualisation. Commercial applications that allow shared visualisation of 3D data are also reviewed.

Kao and Lin (1996) developed a collaborative CAD/CAM system, called Cocadcam, which extends a single location CAD/CAM technology to multi-location application in collaboratively and interactively co-editing CAD geometry at a distance. They also extended the Cocadcam system to remote machining operation at a distance. Unreliable data transmission was tackled by adopting a narrow-band dedicated data transmission link and reducing the data quantity to be transmitted. Their primary focus was on the implementation of the collaborative CAD/CAM environment by instant 3D data exchange, rather than on the issues related to collaborative interactions taking place in shared 3D workspace activity.

Anupam and Bajaj (1994) developed the Shastra, an integrated multimedia collaborative design environment. It provides a geometric and scientific design environment with facilities for geometric design, simulation, visualisation and animation. The underlying structure of the Shastra environment features collaboration, connection, and communication substrates. These substrates are function libraries with well-defined abstract programming interfaces. On top of these substrates, the runtime structure consists of multiple interacting tools. These tools include kernels, session managers, toolkits and services. Kernels maintain the runtime environment, tracking all instances of tools in the distributed system. A session manager maintains a collaborative session, handles connection details, controls interaction and regulates access. Toolkits implement scientific design and manipulation functionality and offers specific services for communication and animation. They presented a scenario that employed the tools provided by the Shastra environment for collaborative geometric engineering design. They conclude that the Shastra provides an enabling infrastructure for rapid prototyping of tools and the runtime environment helps to build multi-user application.

Gisi and Sacchi (1994) developed a prototype system, Co-CAD, that provides a number of features to support synchronous collaboration among a number of mechanical CAD engineers located at different sites. Their prototype system allows users to edit an engineering design concurrently, to customise their local view of a design and to share a common view of a design. The system provides shared pointers, object ownership, and access permissions. Co-CAD is intended to support collaboration that takes place from time to time during design projects. Their assumption about team design activity is that constant collaboration is not necessary throughout the design project. This may be arguable in some stages of the design process. The system has primarily focused on mechanical CAD. The applicability of such collaboration in design practice needs to be investigated further with a more extensive usability study with actual users. Little consideration has been given to the 3D interaction and awareness support in the shared workspace.

Shu and Flowers (1992) presented a multi-user modelling system, Teledesign, which allows people to simultaneously modify a common design in a graphically rich environment. The main focus was to identify and examine groupware interface issues unique to 3D CAD. Experiments were conducted to observe the effects of edit access modes, a simultaneous and a turn-taking mode in two types of collaborative tasks. One of the experimental tasks was the co-operative design of a room, and the other was a well-specified interdependent work such as the collaborative building of a bookcase. They confirmed that a simultaneous mode of edit access is preferred over a turn-taking mode for two-person interactions. Based upon the exploratory study using the system for transferring software knowledge, they implemented the Viewpoint, a pyramid that represents the points of view of different designers and provide method of pointing, to support 3D pointing for collaborative interface. They suggested that allowing designers to have independent points of view optimised parallel activity and assisted the collaboration by feedback from different perspectives. This shows the importance of individual workspace and the seamless interconnection between individual and shared workspace in collaborative 3D modelling activities. The system used a single window display of the workspace. Since most 3D building and editing tasks require multiple windows, a question of whether the result can be extended to more general 3D modelling cases may be raised.

Sakai (Sakai, 1996) described an experiment to explore some of the interaction characteristics of a pointing tool for a shared 3D workspace, which employed a view controlling mechanism, based on spherical co-ordinates, from a linear mouse. A simple instructional task and a collaborative problem solving task were accomplished in common and free view configurations. It was pointed out that a common view configuration was suitable for the instructional task, while free view was better for problem solving tasks. The feasibility of a simple animation based prototype was demonstrated to explore the interaction issues of shared 3D workspace.

Pang and Wittenbrink (1997) developed a system called Cspray to support collaborative 3D volume, surface and flow visualisation. The name comes from a metaphor of a spray-can where the system treats data as invisible entities that must be painted during the visualisation process in order to make them tangible. Cans are filled with paint consisting of smart particles also called 'sparts'. That paint is spread over the data set to highlight certain features in an incremental manner. Some collaborative features that Cspray system provide includes session management, public window and eyecones, private and public spray cans, floor control and data sharing.

Some commercial systems were developed for a conference involving 3D data. These applications are called collaborative 3D model viewers and have origins in single user 3D viewers that enable users to look at a 3D model created in many different CAD formats. The collaborative 3D model viewers provide an additional ability to share the viewing experiences with a number of other users across local network and the Internet. Example systems include ConceptWorks (Figure 2.8) from RealityWave (RealityWave, 2000), OneSpace and FristSpace (Figure 2.9) from CoCreate (a division of HP, CoCreate, 2000), IRIX Annotator (Figure 2.10) from Silicon Graphics (Silicon Graphics, 2000), and eZ (Figure 2.11) from Sigma Design International (Sigma Design International, 2000). The first three concentrate on engineering design, while eZ takes the broader approach of enabling both 2D and 3D collaboration with the goal of supporting both architectural and mechanical engineering users. All these applications emphasise the consideration of non-technical users, such as managers or marketing people, in order to bring real-time 3D viewing beyond CAD users. For fundamental functions of viewing and sharing of 3D models, these tools provide shared camera views, three dimensional cursors, and annotations that allow the participants to discuss specific details of a shared product model. However, except for OneSpace, these systems do not allow users to modify models interactively during discussion and their performance is still quite limited. With OneSpace, users can add markups and edit models; However, any modifications are not saved with the file but are only for visualisation.

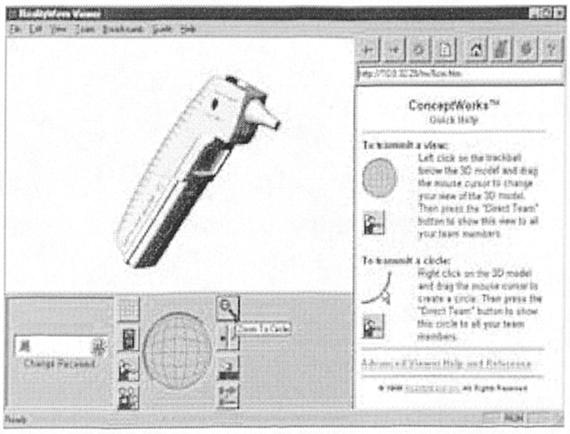


Figure 2.8 Screenshot of ConceptWorks from RealityWave

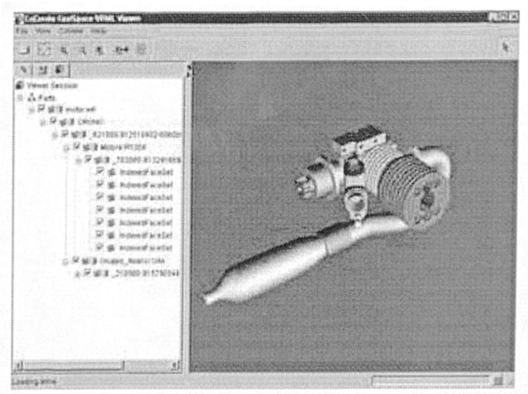


Figure 2.9 Screenshot of OneSpace collaborative engineering solution from CoCreate

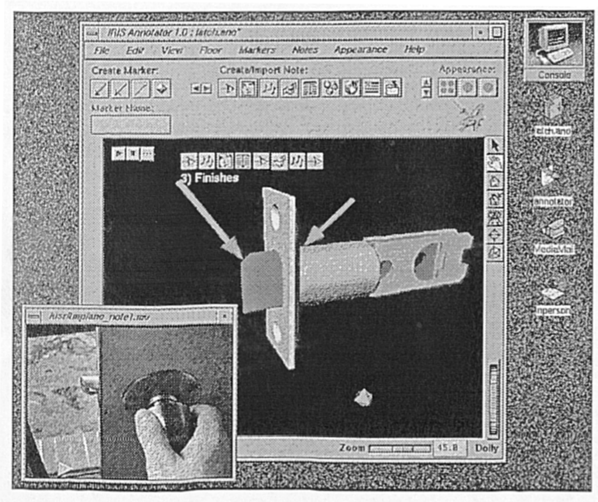


Figure 2.10 Screenshot of IRIX Annotator from Silicon Graphics

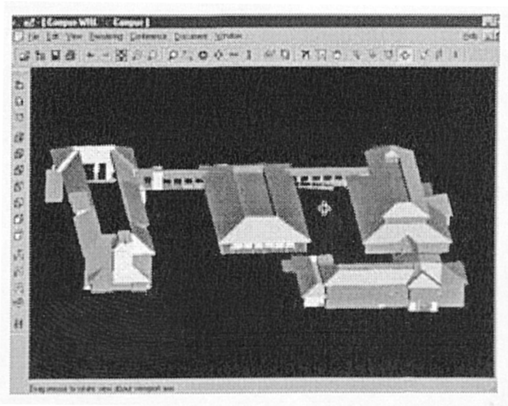


Figure 2.11 Screenshot of eZ collaborative 3D viewer

2.5 Interaction techniques for computer-based 3D workspaces

The 3D interface of many commercial CAD systems is based on two dimensional devices, such as the mouse and screen. Therefore, in a typical computer based 3D workspace applications, there may be user interface problems because the dimension of the interaction device is different from that of the task being undertaken. For example, many commercial applications often provide camera view related features to aid interaction with 3D workspaces. Because a single view does not provide enough location information in 3D, multiple camera views, for example front, top, right and perspective views, are used at the same time to build or modify 3D objects. Consequently, a 3D interface using these views involves many functions to control camera movements, such as zoom, pan, and tumble. The interface of these functions is accomplished through a combination of mouse and key pressing. Sometimes, camera parameters are controlled by slider bars or inputting directly from the keyboard. Therefore 3D operations, which may be simple in the real world, become complicated and unnatural in the virtual 3D environment. To overcome this problem, studies have been carried out to investigate a more natural 3D interaction for computer based 3D work (Sachs et al., 1991; Houde, 1992; De Boo, 1999; Gribnau 1999). In order to know what sort of interaction is appropriate for collaborative 3D modelling workspace, literatures related to the interaction techniques of 3D modelling are reviewed. This review provides the foundation to consider a 3D interaction mechanism used to be extended for collaborative environments.

One of the 3D interface techniques that could be used with CAD systems to provide a more natural way to interact with 3D objects is Virtual Reality(VR). In a virtual reality environment, interaction in three dimensions could be similar to real world scenarios. However, the graphical representation of VR environment is still unrealistic and the equipment required for VR is bulky. Furthermore, sophisticated user interface in 3D workspace activity with VR has not been fully investigated. Therefore, the appropriateness of VR as a 3D interface technique is still being debated. To address some of the limitation of VR, Mixed Reality (MR) or Augmented Reality (AR) has recently drawn the attention of several research groups. There are several advantages to an approach which combines the real world and the virtual model. For example, objects not existing in the real world can be viewed and examined whilst real objects can be augmented by virtual annotations. A selection of interaction techniques using VR and MR follows.

Venolia introduced a direct facile 3D manipulation using a 3D pointing device called a roller mouse (Venolia, 1993). It is a standard one-button mouse with wheels on the front. These wheels vary the cursor-camera separation. When the cursor is hidden by an object, then the object is rendered translucent. When the cursor touches an object, cross-hairs appear within the object. A technique called 'tail-dragging' was also developed to determine the orientation of the cursor, and a technique called 'snap-to' was introduced to help users align objects in both position and orientation. Audio reinforcement was used to accentuate interactions in this interface. Experiments indicated that users found the 3D mouse to be a natural extension of a 2D mouse and were able to control the cursor, and even master the complex interplay between the mouse body, the button and wheels.

Sachs et al. (1991) developed '3-Draw; for 3D interaction when designing forms. The system is based on a two-part input device: including a palette and stylus. The palette and stylus are both tracked with 6-DOF Polhemus 3Space Trackers (Polhemus, 2000). In one hand, a user uses the palette to rotate a wireframe model on the screen. In other hand, a user sketches in 3D using a stylus. Users get a natural feeling of holding the object in their hands while shaping it. They reported that the approach was effective for quickly sketching relatively complex objects.

There have been several studies on a 3D stereo image display system that is augmented by a head tracking device. The synthetic viewpoint of the rendering process can be made to correspond to the actual dynamic physical viewpoint of the user. Deering (1992) raised four issues to achieve accurate high resolution head-tracked stereo display on a workstation CRT; the need for predictive head-tracking, the dynamic optical location of the viewer's eyepoints, physically accurate stereo perspective viewing matrices, and the corrections for the refractive and curvature distortions of glass CRTs.

Experimental results show that if these four issues are addressed it may be possible to achieve sub-centimetre accuracy when the virtual models are superimposed onto the physical world. It may also allow virtual and physical objects to be intermixed.

A similar approach was taken by Arthur et al. using fish task virtual reality (Arthur et al., 1993). They defined fish tank virtual reality as using a standard graphics workstation to achieve real-time display of 3D scenes, using stereoscopic and dynamic head coupled perspective. Several advantages of the fish tank virtual reality were pointed out including the efficiency of resolution, the ability to simulate the effect of depth of field, better stability in the presence of eye movements, and integration in the everyday workspace. In their experiments, users preferred the 'head coupling without stereo' to 'stereo without head coupling'. When both factors are used, better task performance was achieved than using standard display techniques. They also found that task performance is more influenced by the time lag than the frame rate.

Mixed Reality is suggested to address problems caused by traditional Virtual Reality techniques, which separate the user from real world and their traditional tools. Milgram and Kishino (1994) defined Mixed Reality as those in which real world and virtual world objects are presented together on a single display. Several 3D interfaces with single-user Mixed Reality have been developed for computer aided instruction (Feiner et al., 1993), manufacturing (Cruz-Neira et al., 1992) and medical visualisation(Bajura t al., 1992). Billinghust and Kato (1999) suggested the Mixed Reality environment can be used to support local and remote collaboration, by addressing two major issues in CSCW: seamelessness and enhancing reality. They pointed out the advantages of using Mixed Reality for collaborative 3D interfaces. Compared to immersive virtual environments, MR interfaces allow users to refer to notes, diagrams, books and other real objects while viewing virtual images, and users can use familiar real world tools to interact with the images, increasing the intuitiveness of the interface. More importantly, users can see each others' facial expressions, gestures and body languages, increasing the communication bandwidth. Finally not all of the environment needs to be modelled, considerably reducing the graphics rendering requirements. However, MR environments are still dependent upon bulky devices, such as stereoscopic display or head mounted video display devices that superimpose virtual images over a camera captured image, as well as tracking equipment. Many issues regarding display techniques and intermixing the sense of virtual and real worlds remain to be addressed.

Table 2.1. Summary of related studies in the area of design activi ' and sh'ired 2D workspaces

Each study is summarised by the author and date, their research domain, Research methodology used, findings or comments and resulting prototype if one exists.

Research Group	Research Domain	Research Methodology	Findings or Comments	Research Prototype
A1) Delft Workshop, 1996	Design, Engineering, Social Science	Protocol and observational analysis	An overview of the accumulated knowledge on design activity was achieved. Protocol analysis was validated as a research technique for design.	-
A2) Harrison and Minnemen, 1996	Engineering Design	Observational analysis	3D objects are constituents of the activity and communication. They alter the dynamics of interaction in multi-user settings	-
A3) Olson et. al. 1992	Social Science	Protocol analysis	There are similarities in how people spent their time in a variety of design meetings. Design categories that can be used for other kinds of problem solving meetings were developed.	-
A4) Tang, 1989,1991	Engineering Design	Observational and Interaction analysis	Six implications for the design of tools to support shared workspace activity. Importance of gesture and the creative process during collaboration	-
A5) Maher et. al., 1998, 1999	Architectural design	Observational analysis & Case Study	A valuable amount of the semantic information is left undocumented due to the intensive exchange of information via video conferencing Three different collaborative design styles were identified; close-related work, independent work, and work dictated by a leader The concept of a distributed collaborative design environment is examined in a series of Virtual Design Studio projects.	-
B1) Xe oxParc, 1996	Computer Science	Intuitive problem understanding, prototype development and usability study	Ten design principles of object based collaborative 2D graphics application.	Tivoli
B2) Greenberg et al., 1996	Computer Science	Prototype development and usability study	Raising issues of access control, seamless intermixing of workspace actions and functions, private work surface, maintaining shared visual representation for 2D shared drawing activities	Group-Sketch & Group-Draw
B3) Tang and Minneman, 1991	Engineering Design	Prototype development based on observational findings	Three aspects of implications for tools were implemented and evaluated; conveying hand gestures, timing relationships, spatial relationships and concurrent access to the shared space	Video-Draw
B4) Ishii et. al., 1996	Computer Science	Intuitive problem understanding and prototype development	A metaphor of transparent glass board was suggested for a collaborative work. Importance of gaze awareness was proposed. Integration of interpersonal space and shared workspace was achieved.	Team-Work-station, Clear-Board
B5) Scrivener et. al., 1993	Design	Experimental study with prototype system	Feasibility of shared sketchpad was proved for early meetings of design work at a distance. Difference of cultural and work environment influenced the collaborative process.	ROCOCO sketchpad

Table 2.2. Summary of related studies in the area of shared 3D workspaces and 3D interaction techniques

Research Group	Perspective	Research Methodology	Findings or Comments	Research Prototype
C1) Kao and Lin, 1996	Engineering design	Intuitive problem understanding and prototype development	Unreliable data transmission problem was addressed by adopting a narrow band data link and reducing data quantity Instant 3D data exchange was achieved.	Co-CADCAM
C2) Anupam and Bajaj 1994	Computer Science	prototype development and case study	Collaboration connection and communication substrates was suggested. Multiple interacting tools including kernels, session managers, toolkits and services were developed	Shatra
C3) Gisi and Sacchi 1995	Engineering design	Intuitive problem understanding and prototype development	A prototype addressed a group use to edit a design, to customise local view and have shared view, shared pointer, and session management.	Co-CAD
C4) Shu, 1992	Mechanical Engineering	Experimentation with a prototype system	Simultaneous access to the modelling space and independent points of view optimised parallel activity and assisted the collaboration.	Tele-design
C5) Sakai, 1996	Design	Experimentation with multi-media based prototype	Common view configuration is suitable for the instructional task, while free view is better in the problem solving task.	Flying Hands
C6) Pang and Wittenbrink, 1997	Computer and Information Science	Intuitive problem understanding and prototype development	A collaborative 3D surface, volume and flow visualisation system was implemented using a metaphor of spray-can. Various collaborative features including session management, public window and eyecones, private and public spray cans, floor control and data sharing, were examined.	CSpray
C7) Reality-Wave	Mechanical Engineering	Commercial Application	Collaborative 3D model viewer for technical and non-technical users.	Concept-Works
C8) Sigma Design international	Mechanical and Architectural Engineering	Commercial Application	Both 2D and 3D collaboration is supported with the goal of supporting architectural and mechanical users.	eZ
C9) CoCreate	Mechanical Engineering	Commercial Application	Concurrent engineering solutions aiming to support collaborative viewing, markup and modelling.	OneSpace & FirstSpace
D1) Venvolia, 1995	Computer Science and electrical engineering	Case study with a prototype system	3D cursor controlled by a 3D pointing device was proposed. Tail-dragging, snap-to technique. audio reinforcement was suggested to improve 3D interaction.	Roller mouse
D2) Sachs, 1991	Computer Science	Case study with a prototype system	The use of both hands was proposed. Natural feeling of holding and shaping objects was achieved by using palette and styles	3-Draw
D3) Deering, 1995	Computer Science	Intuitive problem understanding and prototype development	Four issues were raised to achieve accurate high resolution head-tracted video: predictive head tracking, dynamic optical location of the viewers eyepoints, physically accurate stereo perspective viewing matrices, correction for the refractive and curvature distortions of glass CRTs	HRVR
D4) Arthur et. al., 1995	Computer Science	Intuitive problem understanding and prototype development	Issues were raised on resolution, the ability to simulate the effect of depth of field, better stability in the presence of eye movements, and being integrated in the everyday workspace. User preference on head coupled representation were illustrated. Lag is more important than frame rates	Fish Tank VR
D5) Billinghust, 1998	Electrical engineering	Intuitive problem understanding and prototype development	More natural support for 3D collaborative work can be achieved and Two major issues in CSCW may be addressed by Mixed Reality: seamlessness and enhancing reality	MR video confer-encing System etc.

2.6 Summary of the review and research gap

Studies relating to collaborative design environments have been reviewed from the broad perspective to the specific focus on real-time collaborative 3D CAD. A summary of the previous studies is presented in Table 2.1 and Table 2.2. Each study is summarised by author and date, their research domain, research methodology used, findings or comments and resulting prototype if one exists.

Firstly, the review of studies on team design activity provides a social and psychological foundation. Some studies in this area, for example papers from the Delft workshop (Cross et al., 1996), have attempted to systematically analyse team design activities to improve understanding of the design process. In particular, as a growing number of investigations have emerged for building tools to be used by multiple designers (Tang, 1991; Olson et al., 1992), computer support for collaborative design has become a major area of research in the design domain (Maher et al., 1999). Researchers in this area often employ protocol analysis techniques for analysing design activities. Although protocol analysis has been demonstrated as a useful methodology for understanding design activities (Cross et al., 1996), it must be assisted by other qualitative approaches in order to capture all aspects of design activities. Some researchers emphasize the importance of qualitative analysis (e.g. Tang, 1991) since they consider that methodologies of analysing design activities and processes can not be clearly established. Researchers in this area generally accept that our understanding of team design activities is incomplete because of the complexity and variety of design activities. More empirical studies are required to improve the understanding of the context of collaborative design environments. In particular, further studies are needed to apply our understanding of design activities to the investigation of tools to support the activities and to produce better design outcome more efficiently.

Secondly, many studies have been carried out with regard to shared 2D workspace in the area of CSCW. Particularly, there have been many investigations for shared drawing and writing, although most studies targeted a general group task. These studies have mostly originated from a computer science perspective and tend to focus on technical issues of real-time distributed applications rather than on user interface issues in a particular collaborative environment. Some researchers employ design implications drawn from design activity studies (Tang, 1991) in order to develop research prototypes (e.g. Greenberg et al., 1996). In order to investigate ways of supporting more sophisticated and complex professional collaborative activities, however, insights are required from the domain in which the collaborative activities take place. In order to support professional collaborative design practices, design research need to provide further insights on design tools and environments from a user's perspective.

Thirdly, studies on shared 3D workspaces have also been reviewed. In collaborative CAD and scientific visualisation, 3D models are the main focus and important elements of activities. Among the studies in this area, Cspray by Pang and Wittenbrink (1997), Teledesign by Shu and Flowers (1992) and Co-CAD by Gisi and Sacchi (1994) have some aspects in common with this investigation in terms of their research goals, which is to support real-time collaborative 3D activities. However, these focused on particular problems of the collaborative task or had different target user groups from designers involved in a product development process. The application domains of these works range from architectural design and mechanical engineering to scientific visualisation. Therefore, problems may arise when one extends the research findings or prototypes to real-time 3D CAD for designers. In addition, the analytic evaluation of the prototypes has not been fully conducted. Detailed interaction analysis with new tools is therefore required to investigate a suitable direction for the development of the next generation of real-time collaborative 3D design tools. It can help address 3D specific user interface issues and the relationship of 3D workspace to other collaborative technologies.

Finally, a variety of 3D interaction techniques have been examined to consider a 3D interaction mechanism to be extended for collaborative environments. That includes the conventional screen and mouse-based interface (Venolia, 1993), 3D interaction techniques using 6 DOF devices (Sachs et al. 1991), virtual reality (Dering, 1992; Artur et al., 1993), and augmented and mixed reality techniques (Milgram and Kishino, 1994; Billinghust and Kato, 1999). Although these studies have suggested a variety of new techniques for natural 3D interface, further consideration should be given to ways of incorporating these interaction techniques in a real-time collaborative design environment to facilitate design activities as well as co-ordination processes.

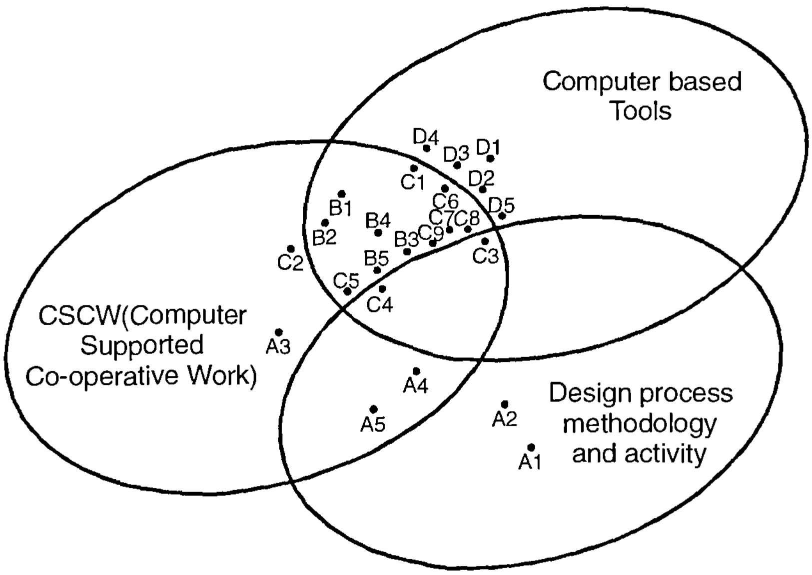


Figure 2.12 Research domains of related studies: The identification code of the research (e.g. A1) is presented in the Table 2.1 and Table 2.2

The research domains of the related studies may be mapped into the scope of the thesis illustrated in Figure 1.1. Figure 2.12 shows where each study reviewed is located in the different research domains. As shown in the figure, a number of investigations have been carried out in the area of CSCW and computer-based tools. However, comparatively little investigation has been carried out in the overlapping research domain integrating the related areas.

The research reviewed in this chapter comes from a variety of fields, including computer science, social science, engineering design, architectural design and engineering design. Despite this diversity, there is a deficiency of studies on computer support for design activities that are originated from user's perspective. In fact, most of the tools for shared 2D/3D workspace activities are initially targeted at the general collaboration tools, so they have overlooked some unique aspects of design activities, such as 3D CAD.

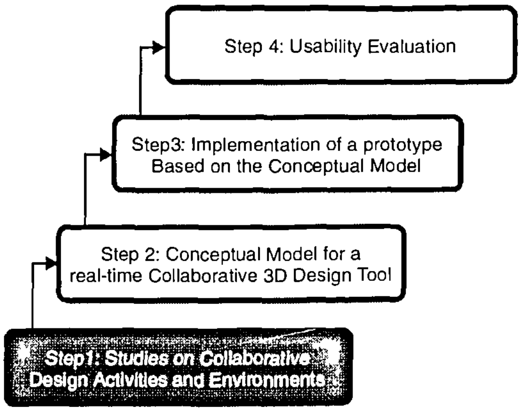
Most of the studies reviewed for shared workspace focused on specific issues to be addressed for the development of general synchronous collaborative applications. They have largely focused on collaborative document editing, drawing and authoring activities. Comparatively little study has been directed towards tools supporting design activities around the 3D CAD workspace. In particular, the real-time support of shared 3D workspace activities has not been fully explored. Although some researchers have looked at real-time shared 3D modelling or viewing support, they mainly focused on fundamental infrastructures (Anupam and Bajaj, 1994) or approach the collaboration issues from an engineering perspective (Gisi and Sacchi, 1995; CoCreate, 1999). As indicated above, none of the previous studies provided an analytic evaluation of real time collaborative 3D design tools in an environment involving real users. Thus, the usability and usefulness of real-time collaborative 3D CAD systems for designers have not been fully addressed.

Some studies of shared workspaces have focused on collaborative 2D drawing or writing tools. 3D visualisation and construction plays an important role in the design process because it allows designers to advance ideas, developing an initial drawing. The tools to support 3D become important after initial idea generation. However, relatively little consideration has been given to collaborative 3D design tools. The previous investigations reviewed tend to focus on ways to support a conference that involves 3D data. The activities of real-time collaborative construction or modification of 3D models using CAD systems have not received full attention.

The review of related works provided a foundation for the technical and methodological approaches of this thesis. The review also illustrated that there is a deficiency of studies in the area of shared 3D workspace for design activities. In particular, it was found that there is a research gap in real-time collaborative 3D design tools to support synchronous shared 3D workspace activities. The review also highlights that an approach from designers' perspective and user study would be invaluable for these new design tools.

Chapter 3

Understanding Team Design Processes and the Impact of Collaborative Technologies



3.1 Introduction

This chapter presents a preliminary investigation of collaborative design environments before focusing on real-time collaborative 3D CAD issues. As pointed out in Chapter 2, the development of collaborative design tools should be based on a sound understanding of team design activities. By studying how designers work together in various situations, one can seek the new opportunities offered by collaborative design tools and environments.

A typical team design approach, either for a distributed or for a co-located collaborative design project, can be characterised by a series of meetings in the critical phases of design project life cycle. Although a collaboration takes place using other means of communication, such as telephone, fax and email, throughout the design process, face-to-face meetings play an important role for collaborative activities, such as brainstorming, decision making, information sharing, project management and co-working. By capturing and analysing the face-to-face meetings in a team design project, it is possible to gain a broad perspective on important features of the collaborative design environment.

On the other hand, existing synchronous collaborative technologies allow designers to have real-time electronic meetings, when team designers are not co-located. These collaborative technologies include desktop video and audio conferencing systems, application sharing software tools and shared drawing tools. Investigation into the impact of such collaborative technologies can guide us in developing new collaborative design methodology and tools.

In order to investigate team design activities and the impact of collaborative technologies in design processes, an experimental team design project was carried out and observed in two different conditions. In the first experimental condition, distributed designers accomplished a design project by a series of face-to-face design meetings. The designers in the second accomplished the same design project by a series of electronic meetings equipped with a set of collaborative technologies, including video conferencing and shared application tools. In the following sections, the details of goals, methods and conditions of the experimental team design projects are explained and an observational and comparative analysis, in terms of the design outcome of the projects, designers activities, their collaborative interactions, and the impact of collaborative technologies are presented. The implications for new tools in each stage of design project and the connection to the later chapters are also discussed.

3.2 Objectives of the preliminary investigation

Studies have been carried out to improve the understanding of team design activities and feasibility of new communication technologies (Cross et al., 1996, Scrivener et al. 1993; Maher et al., 1998). Short design sessions in an experimental setting have been frequently used for these studies. A design process however involves many different stages and designers' activities and the necessary tools are different in each stage. It is necessary to identify resources and hindrances in many different phases of a team design project. The preliminary study partially addresses this issue through experimental team design projects.

Considering the immature state of techniques for studying design process and activities, it is generally accepted that a qualitative and exploratory approach is more appropriate to study design activities because of their creative and extemporaneous nature (Tang, 1991). It is particularly difficult to conduct a quantitative analysis because studying actual design projects involves several people over longer periods of time and the data is based on more indirect reporting, through recollection or interviewing. Therefore, in order to conduct effective exploration on overall team design processes and activities, an observational approach is used as a major analysis method. The preliminary study has been mainly intended to clarify research problems of collaborative design environments which are raised in Chapter 1. An in-depth analysis of team design activities was not intended here that may require more accurate experimentation and analysis, or realistic field study involving many groups of designers. The major goals of the preliminary investigation of collaborative design environments are:

- To understand the process and activities of team design,
- To compare a conventional collaborative design environment and a computer mediated collaborative design environment.,
- To identify the impact and problems of existing collaborative technologies in real-time collaborative design sessions,
- To identify the kinds of collaborative technologies necessary in each stage of design process,
- To clarify research issues for collaborative design environments, in particular, issues to support real-time collaborative 3D design activities.

3.3 Procedure

A hypothetical collaborative design project was carried out by two designer groups in different environments where designers worked as if they were physically distributed. The first group accomplished the project by having conventional face-to-face meetings while the second group interacted via electronic meetings.

In order to observe all phases of a team design process, participants were asked to finish their design project within three days and to have only four meetings throughout the design project. The number of the meetings was confined to four because participants accomplishing a collaborative design project in a distributed situation would probably not have frequent meetings. It is also expected that four meetings in critical phases of the design process could advance the design task given in the experiment.

From a survey of the experience of design practice and literacy on computer based design tools, six participants were recruited from students in the Department of Design at Brunel University, UK. They were randomly grouped into two three-person teams. The first design team had four face-to-face meetings in a meeting room, which was equipped with a large discussion table, whiteboard, pens and paper. A computer was also supplied to provide general computer based design tools, such as desktop publishing, image processing, 2D illustration, 3D rendering and CAD applications.

The second team accomplished the same design project, having four electronic meetings. Each participant was located in a different room during the electronic meetings and they did not meet each other face-to-face until the end of the project. Each room was equipped with a computer connected via a LAN for desktop conferencing. Two additional computers were configured as servers: one for the video conferencing and the other for application sharing. The computers in each room were connected to the servers as client systems. General computer based design applications were installed in the shared application server. A telephone conferencing system incorporating headsets was used for voice communication. Telephone conferencing was used instead of computer based audio conferencing to reduce network traffic and increase the performance of the computer system during the electronic meetings. Details of hardware and software used in the two experimental conditions are illustrated in Table 3.1.

Table 3.1 : Details of hardware and software used in the two experimental conditions

Session ID	Details
I: Distributed team design project with face-to-face meetings	One meeting room equipped with a large discussion table, pen and paper, whiteboard and one Pentium PC
II: Distributed team design project with electronic meetings	Server systems • Video conferencing server hardware: Silicon Graphics, • Video conferencing server software: CUSeeMe reflector • Application sharing hardware: Pentium PC • Application sharing software: Timbuktu Pro Client Systems • Hardware: Pentium PC which QuickCam • Video conferencing : CUseeme (Version 2.1) • Shared Remote control : Timbuktu Pro • Shared whiteboard : Teamroom (Roseman and Greenberg, 1996a; Teamwave, 2000) • Audio Conferencing : 'Monarch 120B' telephone system and headset phones

3.4 Design task

Considering that the student designers had an overall understanding of the design process, but did not have the practical experience required to solve complicated design problems, a design of an emergency car torch was chosen as a design task. This task was defined in order to ensure that it could be completed in a convenient time scale and it was within the capabilities of the participating designers. The design task given to the participants for the experiment is shown in Figure 3.1.

Design Assignment : Car Torch

Namil Corporation is a small manufacturing and assembly company that specialises in car accessory products. They are renowned for their outstanding and unique design of car accessory products. With a growth in manufacturing capacity, they are trying to find a new item to extend their product range.

One possible product is a lighting device required by car drivers who have to check car bodies or engine components at night, especially in an emergency situation. This may also be used to inform other drivers on the road of their presence. It is also expected to be used as an auxiliary light for drivers and passengers inside a car. H-Car, one of the most successful car makers in Europe and a close partner of Namil Co., is also interested in this product and intend to invest partially in this project.

Because Namil Co. thinks that this project will influence significantly on the future co-operation with H-Car, the representatives of Namil Co. came to your design group and asked you to develop a concept design for the product. In three days there will be a meeting with the Namil Co. and the managing director of H-Car. Before then Namil Co. needs to have a clear idea of the kind of the product, and a detailed proposal.

Namil Co. have defined some requirements of the product to be :

- Usable as an emergency torch
- Compact size
- Use as an auxiliary interior light (e.g. reading a map at night)

For the submission of the final design, you are asked to produce a design proposal on A4 boards in which you need to explain design specification, and include presentation
Drawings and scale drawings with dimensions. Any models or design sketch work will also be collected.

Figure 3.1 The task used in the experimental team design project

Participating designers took three whole days to accomplish the design assignment. Meetings were evenly distributed throughout the three days and a broad meeting agenda was provided to give an indication of what was required of the participants in each phase. In the introductory meeting, on the first day, the design brief and the project schedule were explained to the participants. On the same day they had the first meeting. The suggested agenda for the first meeting was a discussion about the design problems. In the next day, they had two meetings, one in the morning and the other in the late afternoon. The topic of the second meeting was suggested as idea generation and selection of alternatives. The agenda for the third meeting was detail refinement of the selected design. The final meeting was held in the afternoon of the last day. At the end of the final meeting, they were required to prepare a presentation of their design proposal.

Designers in the electronic meeting environment also followed the same schedule. Additionally, during the introductory session, they were introduced to the collaborative tools used in the electronic meetings. They accomplished a simple logo design exercise using the tools. Table 3.2 shows the project schedule and suggested meeting agenda.

Table 3.2 Project timetable and suggested agenda of the meetings

Meeting ID	Suggested agenda	Meeting Time
Introductory Meeting	-	Day1 09:00
1st Meeting	Understanding design problems	Day1 14:00
2nd Meeting	Idea generation and selection	Day2 09:00
3rd Meeting	Improvement of details	Day2 17:00
4th Meeting	Finalisation	Day3 15:00

3.5 Data collection

The four face-to-face meetings for the first team project were videotaped from three different angles; a video of the whole room, desktop view, and computer screen. Video recordings captured most of the designers' activities and interactions in the meeting room. Personal sketchbooks illustrating their individual design progress and drawings produced during the meetings were collected and coded with the time of creation. The four electronic meetings of the second team were also videotaped with audio. Since participants were distributed in three different rooms, three video cameras were installed to capture the individual participant's activities. Their video conferencing view was recorded along with the telephone conversation. Parts of the audio recordings were transcribed. Figure 3.2 shows the sample video recording used for the review. After the final meeting, interviews and questionnaires about each team's project were completed.

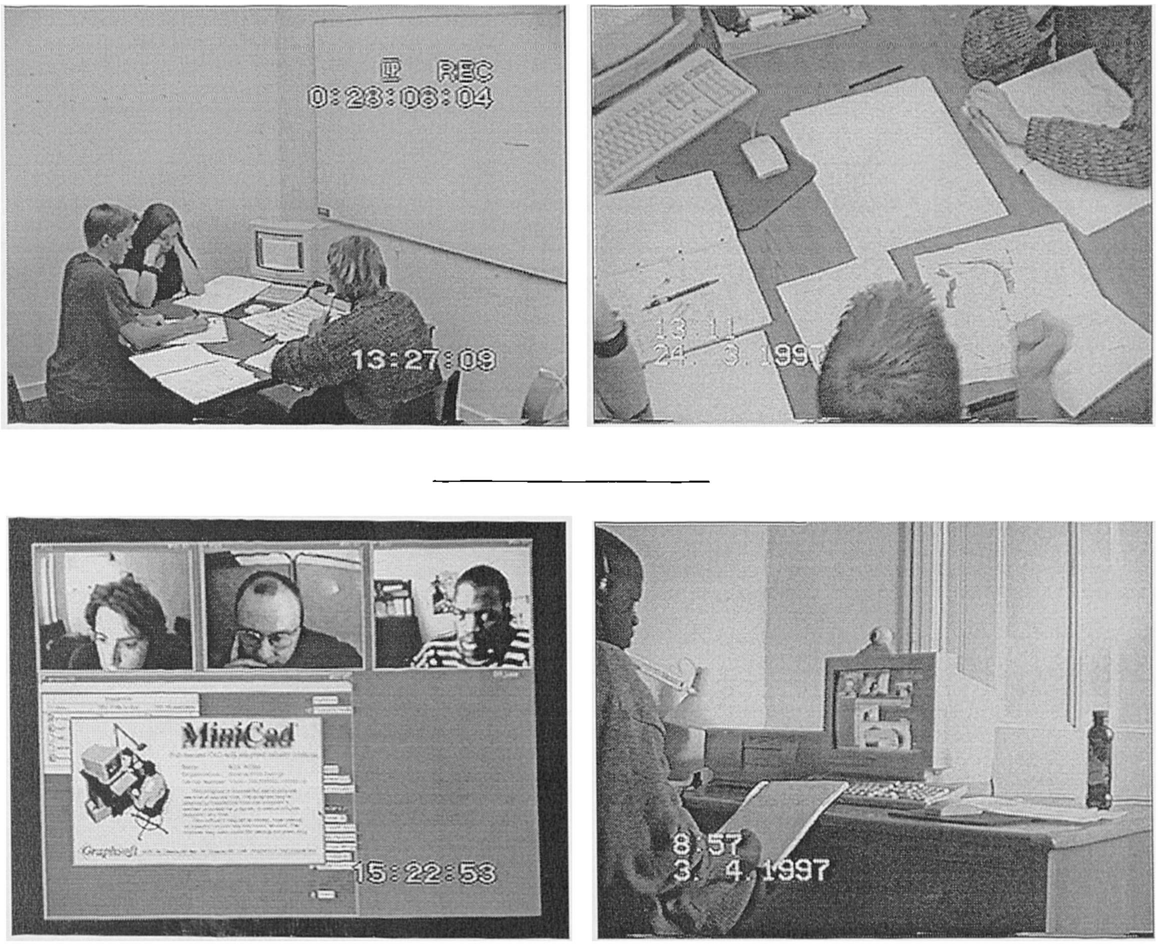


Figure 3.2 Configuration of face-to-face and electronic meeting sessions. Top figures show the room configuration (left) and a desktop view (right) of the face-to-face sessions. Bottom figures show an individual computer screen (left) and room configuration (right) of the electronic meeting.

3.6 Results

3.6.1 Quantitative results

In order to investigate the two conditions, three aspects of the team design project are first analysed quantitatively: the meeting time, design outcome, and the participants' response to the process, outcome and tools used.

3.6.1.1 Meeting time

The average length of the face-to-face meetings was 71 minutes, while that of electronic meetings was 53 minutes. The meeting time in the early phases was longer in both conditions because the groups spent a long time discussing various issues and identifying design problems. Most of meetings took less than 100 minutes, although there was no restriction on meeting time. The final meetings were relatively short since they were only used for presenting the outcome. It was expected that the electronic meetings would take longer because they had to make more effort to communicate. However, it appeared that the difficulties of communication forced designers in the electronic meetings to finish the meetings comparatively quickly. Communication difficulties were frequently expressed by the participants during the electronic meetings. Table 3.3 shows the time length for the meeting at each phase of the team design projects and Figure 3.3 shows the comparison of meeting time between two teams.

Table 3.3 Time taken for meetings in both projects. (Unit: Minutes)

Project	**Phase 1 Meeting**	**Phase 2 Meeting**	**Phase 3 Meeting**	**Phase 4 Meeting**	**Average**
with face-to-face meetings	94	75	70	46	71
with electronic meetings	60	55	55	40	53
Average at each Phase	77	65	63.5	43	62

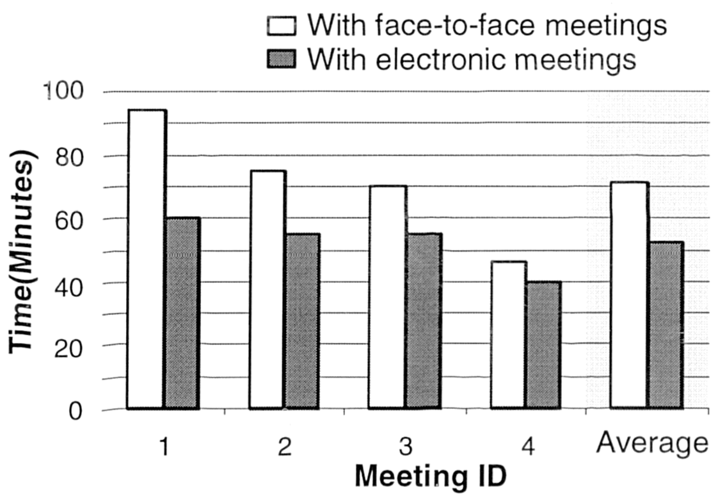


Figure 3.3 Comparison of meeting time between two projects

3.6.1.2 Design outcome

The design outcomes of each team were evaluated by five professional industrial designers. The criteria for the evaluation were aesthetics, user interface, originality of idea and the way that the final proposals were presented. Table 3.4 shows the average marks of interval scores from the questionnaire. Since quality of outcome is also influenced by many other factors, such as abilities of individuals and the composition of the team, further investigation needs to be carried out to generalise the result of the assessment. However, the team in face-to-face meetings seem to produce better outcome than with the team with electronic meetings in general, while there is comparatively little difference in the way that they presented the final proposal. This indicates that groups using electronic meetings can perform equally well in some stages of team design processes, particularly in the preparation of the final presentation. Design outcome and assessment details are provided in Appendix A.

Table 3.4 Assessment result of the quality of design outcome (1: lowest, 7:heighest)

Design Team	Aesthetic aspect	Consideration of user interface	Originality of design	Quality of final proposal
With face-to-face meetings	5.6	4.6	5.0	3.6
With electronic Meeting	2.8	2.8	3.0	3.4

3.6.1.3 Participant response

Both teams showed a high level of satisfaction about the outcome and the way that they worked together. Although the responses of the face-to-face condition were slightly more positive, there was no marked difference between the two conditions (Table 3.5). Participants in the computer-mediated condition expressed some difficulties using the collaborative technologies. In particular they observed the quality of video and audio was not good, and that these interfered with their ability to interact with the rest of the team. Nevertheless, they reported that the collaborative technologies allowed them to work together without face-to-face contact.

Table 3.5 User response about the process and the outcome of the project

Q1: Are you satisfied with the final design output?

Q2: Are you satisfied with the way of team working in this project?

Q3: Are you satisfied with your role in the group work?

Q4: Are you satisfied with the group members?

Design Team	Q1		Q2		Q3		Q4	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
With Face-to-face meetings	1.0	0	1.0	0	1.3	0.6	1.3	0.6
With electronic meetings	1.3	0.6	1.7	0.6	2.0	0	2.3	0.6

The figures are interval scores, representing 1 the most positive and 5 the most negative

3.6.2 Observation of team design processes

The designers in the face-to-face condition started with a discussion about the components and features of the torch then exchanged concepts and sketches. At the end of the second meeting, they were able to decide the final design except for a few details. The third meeting was spent clarifying the specification document. In the final meeting, they confirmed the final drawing and specification. The designers in the computer-mediated condition had a similar discussion in the first meeting. They then exchanged visuals in the second and third meeting. They decided on one of the ideas proposed but did not have a full discussion of the details. In addition, the work of producing the final proposal was accomplished individually. The final meeting was used for integrating work prepared by individuals. Both teams accomplished the project in a similar way but the progress of the electronic meeting condition was delayed and decisions sometimes made without full agreement. Once a decision had been made, the designers in the electronic meeting condition tended to work more individually in producing a final proposal. The details of the observation on each phase of the team design processes and activities are presented next.

3.6.2.1 Session 1: distributed team design project with face-to-face meetings.

1) Phase 1

After the introductory meeting, three designers (Designer T, L, and G) had about 4 hours of individual working time before the first team meeting. All three designers engaged in activities of identifying design problems and understanding the design task. They listed product features and functional requirements of the car torch, based upon the design brief. Each designer produced various design ideas before the first meeting. As shown in Figure 3.4, some designers focused on identifying design problems described on the design assignment sheet, while others developed a fairly detailed design concept based upon their own interpretation. For example, Designer T focused on identification of what was described in the brief and the general goals of this project. He listed some of the requirements of the product and its target users and various situations where the product would be used. Designer L divided the car torch into sub units such as the light, switch, and battery components. She then considered some details such as basic shape and arrangements of possible components. Designer G developed the design concept to a greater level of detail than the other two. He quickly made his own decisions on product features to be incorporated and functional requirements. Then he produced a soft mock-up of design concepts for the first meeting based upon his initial decisions. Different approaches and progress of individuals had to be co-ordinated and adjusted in the first meeting to create a team strategy.

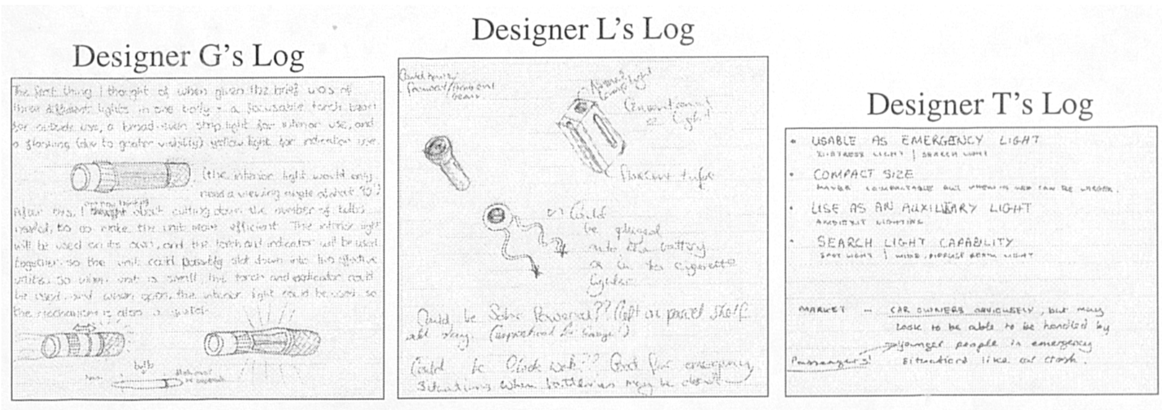


Figure 3.4 Outcome of individual designers before the first meeting. The sketches show different approaches that had to be adjusted for the team project.

In the first meeting, they exchanged their interpretation of the design brief and explained what they thought were important product features and requirements. The directions of the visual and aesthetic aspects of the product were also considered. Exchanging their views on the design problems were largely done by orderly presentation of all members using the prepared listings and drawings. Substantial verbal communication took place.

In addition to exchanging the individual views, new ideas were also proposed during the meeting. It appeared that they sometimes came to a new idea while they were explaining ideas or clarifying what other members implied. Most ideas were described verbally and in less detail in this phase. Designers shared individual concepts and narrowed down alternatives among suggested concepts. Later, part of the meeting was dedicated for the management of the project: things to do and the schedule were discussed. They agreed that the next phase would focus on idea generation.

Figure 3.5 shows some sample sketches and designs produced during the meeting. Various necessary product features and components were discussed. Designers developed their design to use a single light source with multi-functionality, such as spot, ambient and emergency lights. User interfaces of methods of holding and storing were discussed.

Figure 3.5 Sketches produced collaboratively during the meeting

2) Phase 2

In the second phase, designers were actively engaged in producing visual representations of product ideas. They started to use computer based tools for creating 3D renderings of design concepts. Since they narrowed the directions of design and agreed on some fundamental features for the components of the torch at the first phase, the diversity of designers' approaches was reduced. Consequently, they started to build a common understanding and developed problem solving methods. As shown in Figure 3.6, idea sketches from the three designers were increasingly convergent. They focused on shape development and solutions for detail interfaces. The listing of product features discussed and agreed during the first meeting was used as a principle guideline for detail development.

Designer G's Log Designer L's Log Designer T's Log

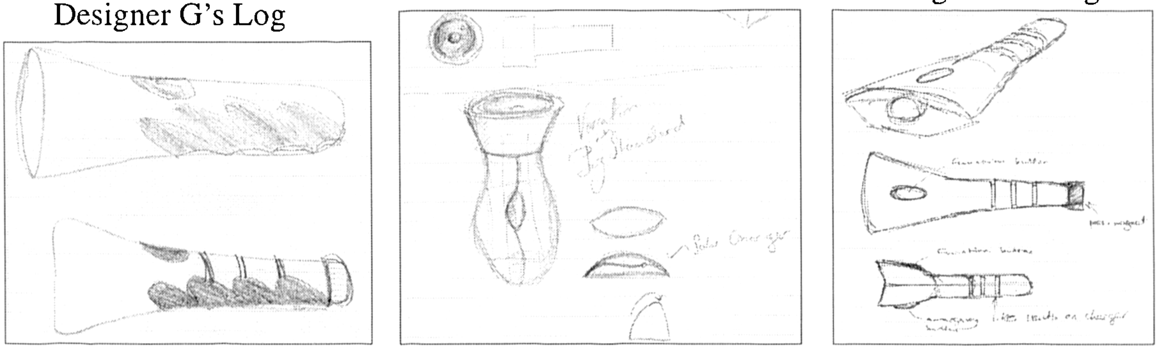


Figure 3.6 Reduced diversity in idea sketches at the second phase

They first focused on a solution of switches and controls. The separation of battery charger unit and main torch unit was proposed and decided as their main design feature. The shape of the torch incorporated a modern look by using an oval shape for the lens and circular shape at the handle. They almost reached a final solution in this phase except for a few minor details.

The presentation of their ideas using simple sketches and drawings was one of the activities frequently undertaken by designers in this phase. All three designers prepared presentations of the product to explain their ideas about details and shapes. One designer used a 3D modelling application as his main representation tool. In the early part of the meeting, they presented their individual solutions one by one. While one person presented his/her ideas, the others constructively analysed various aspects. The order of presentation seemed to influence the decision making process; ideas presented early drew more attention and were used more frequently as a reference for discussion throughout the meeting.

Social aspects of teamwork became apparent as they spent more time working together. For example, it seemed that designer T played a managerial role by making important decisions, negotiating and scheduling the project plan. On the other hand, designer G became less active after the first meeting when his initial idea was not thoroughly considered by other members. It also seemed that personal characteristics and relationships between members influenced the process of team design as well. Designer T and L talked a lot and were involved more in the discussion, while Designer G spent a long time just listening. As a result, designer T and L had a larger influence on decision making and the way they would work throughout the remaining project. Different levels of participation with the project were also observed. For example, the author of a particular idea, which was central to the design, became more active, while one who was not supportive to the idea became less active in the project. The overall atmosphere became more informal and relaxed towards the later part of the meeting. Sometimes, decision making became more complex when members' opinions were divided. This was particularly true when the issues concerned something that was not quantifiable, such as shapes or colours.

Design activities became more dynamic in the second meeting. While designers were involved in verbal discussion most of the time during the first meeting, at this meeting they were involved in various activities simultaneously. For example, while two designers discussed a design problem, the other was drawing and producing other design solutions. Parallel activities were frequently observed. Time planning and project management tasks were accomplished at the end of the meeting. Because of time pressure, they decided to progress into the next phase in which they were required to produce specifications and a design proposal.

3) Phase 3

Designer L produced a manual rendering according to the agreed product shapes and features, while designer T refined the 3D computer rendering. On the other hand, designer G tried different options since there was disagreement on shapes and colour. His new option was however similar to that agreed by the other two designers, because it was too late to make significant changes.

In the third meeting, there were more refined visual representations of product shapes and features. Some details, such as switch mechanisms and general shapes, which had been modified in the individual session, were discussed and evaluated together. They reached the final decision in the early part of the third meeting. Once they agreed on the final design, they became actively involved in producing a design proposal. The requirements for the final proposal were considered and ways to work together to produce it were discussed. It seemed that the tools they used made significant impacts on ways of working at this stage. They understood that technical specification is a part of the outcome and realised that production of the specification can be accomplished together. The rest of the meeting was spent on writing the technical specification of the product collaboratively. In this collaborative production process, ideas and contents were clarified quickly and refined. When the technical specification was completed, they divided other production work for the presentation.

In terms of design development, they clarified what was agreed and wrote this into the technical specification. The final design had to be provided by realistic simulation so that all the team members could share a common understanding of the final result. Since tools for collaborative modification were not available, the detail refinement had to be accomplished by individual members. After deciding requirements and directions collaboratively, a significant change on the direction could not be made individually. This meant that a team design project had less flexibility in trying different options for the solution. As the team design project produces less flexibility in decision making, extra attention should be paid to the initial design phase.

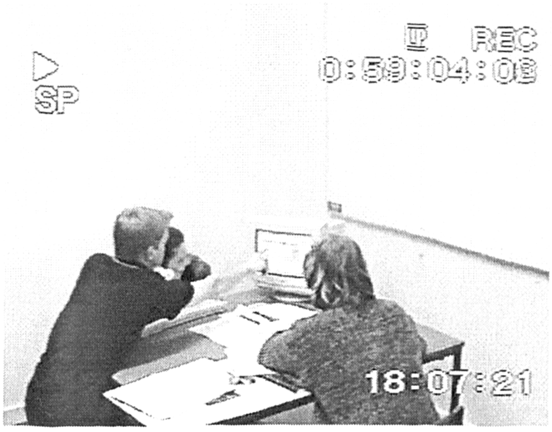


Figure 3.7 Collaborative working with the PC for producing technical specification

4) Phase 4

According to the division of tasks decided in the third meeting, each designer worked individually on the tasks allocated. Designer T modified and completed the final computer 3D renderings. Designer L refined dimension drawings based upon her manual rendering. Designer G did not make a significant contribution to the production of the presentation. In the final meeting, they mainly discussed how to present their design proposal and prepared the final visualisation as a computer rendering and dimensioned drawings. They also included the technical specification as an electronic document.

Because of the distributed situation, it was difficult to share work. The difficulty of task distribution and the lack of tools supporting parallel collaboration are possible reasons why the collaborative activities in this phase were carried out on an individual basis.

3.6.2.2 Session 2: distributed team design project with electronic meetings

1) Phase 1

As part of the introductory session, designers of the second team undertook a simple graphics task in order to familiarise themselves with the collaborative tools used in this study. It was observed that most of these tools presented little difficulty for the designers. Before the first electronic meeting, they focused on identifying design requirements and existing problems, as had the first team. They also investigated the mechanisms used in other torches and similar products.

During the first meeting, most of the time was spent discussing product features and initial concepts by exchanging ideas without using any 'visuals'. The design brief was referred to from time to time in order to clarify direction. After half an hour of discussion, they started to use the desktop conferencing tools. It appeared that sharing what they had drawn or listed during the meeting was difficult and unnatural because they had to use small camera windows. The last part of the first meeting was dedicated to project management: they discussed what they should do in preparation for the next meeting. It was noted that the meeting finished relatively early, even though there were many features to be considered and much information to be shared. They also experienced difficulties in communication due to the small screen and lack of colour images.

The main features the team discussed in the first electronic meeting included component details and operation features, such as battery shape and size, number of bulbs, magnetic attachment, flashing option for emergency situation and being able to use as a reading light. After an initial discussion, the main points mentioned were shared by all the team members.

They quickly became competent in the use of the communication tools, the camera being the main means of visual communication. However, they used the camera for sharing ideas developed during the meeting rather than for watching other participants. The process of sharing sketches and idea representations with the digital camera caused many problems. The picture quality was poor for detailed drawings or text because of low resolution and image quality. The position of the camera on the top of the computer

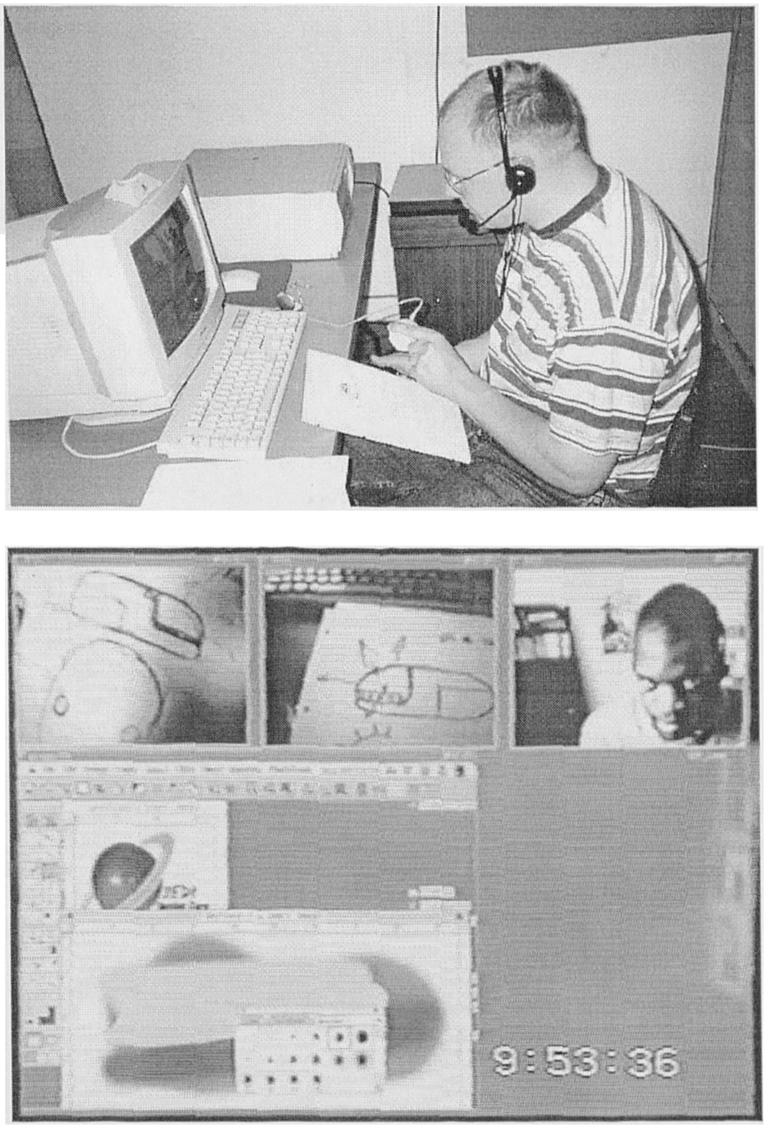


Figure 3.8 The use of camera in the electronic meetings: The way in which they hold camera toward the sketches for explanation (Top). A screen shot of what participants see during idea exchange (Bottom).

monitor was a problem during communication. Sketches could not be shown easily during the discussion because the camera was directed toward their faces. To address this problem, they would hold the camera towards the sketches to aid explanation to other members. Physical pointers, such as a pen or fingers, were used to point on the sketches. They frequently complained about the size, the operation of the video camera and the picture quality during the meeting. The camera's auto-adjustment feature to the change of brightness caused another technical problem when it was moved to show their sketches. Nevertheless, the video camera seemed to play an important role in the discussion. They used the shared application less frequently than as expected. It was used by them in discussion of particular visual aspects, such as colour or dimension. It was also observed that their activities seemed to be more relaxed as they worked alone in the office. Figure 3.8 shows people using a digital camera for idea exchange and the screen shot of what participants saw while others moving the camera.

2) Phase 2

They investigated more detailed design concepts and sketches at this phase. Idea sketches were presented one by one. Discussion of specific detail of the visual aspects of concepts was difficult. Therefore, the number of concepts discussed during the meeting was relatively small. Nevertheless, they reached decisions on the general shape and features of the design relatively quickly.

Communication difficulties seemed to cause a premature final decision, although the proposal could have been refined further. In the following excerpt from the electronic meeting, the participants repeatedly express communication difficulties.

J there and, where is my pen … <u>this is not very easy</u>

G I know what you mean

J and like uh that's the sort of band going over the top, ya and like the lens is on a sort of constantly amber lens would have the like a, it would come out in sections and rotate round so it went all the way round there

G yea

J Can you follow that?

G I think so

I   not quite ...   not quite
J   quite difficult to explain, especially when I can't ...

Once they appreciated the difficulties, the meeting was mainly used for essential co-ordination activities, such as drawing an agreement on the concepts and clarifying the requirements etc. The designers often worked individually during the meeting.

Shared application tools were not used much during the meeting. The slow response of the user interface, for example selecting menus or mouse movement, considerably limited the usefulness of the shared application window. The latter was largely used for simple operations such as showing colour and opening prepared computer images. Difficulty in using shared application tools could be because the existing tools provided in the application sharing environment were not appropriate in a situation where concepts needed to be quickly visualised and shared with other team members. Most of the tools provided in the shared application were designed to be used in the more detailed stage of the design process. In addition, these tools were only designed for a single user environment. Therefore, designers did not have any information about who is controlling and using the shared application. Users' unfamiliarity with the software also made them steer away from the shared application tools.

3) Phase 3

As they approached the later phase of the project, they were more concerned with what had to be produced at the end rather than refinement of design concepts. Therefore, little consideration was given to new design concepts proposed during this phase. Instead, they all unconsciously agreed that ideas proposed at the last phase would be the final concept to be presented. They reviewed project schedules and what had to be produced before the end of the project.

Computer renderings were prepared and displayed in the image editing application running under the shared application environment. Such operations require more screen space and a bigger screen. A CAD application was also used to determine rough dimensions. However, the designers had difficulties in using it because the interface was unfamiliar. Although they almost reached a final decision, they could not share exactly the same final drawings. Therefore, frequent confirmation and clarification of the shape was observed. Overall dimensions as well as shape were agreed but many details had to be decided by individual designers. The initial draft of the specification document was used for confirmation. In preparing a technical specification, a word processing application in 'the shared application tool' was used. They divided the tasks into computer rendering, documenting specification and creating a dimensioned drawing.

4) Phase 4

The design team presented individually and collaboratively completed work. Use of the shared application was important at this phase, in order to share finished drawings and CAD model. Sometimes the team needed to co-work in real-time through a shared application environment. Since all members were able to see the shared application, they could help each other to solve operational problems such as opening and transferring files. One designer showed the final version of his computer rendering, in which the colour and the texture of the product were modified. Changes to a few details in the final stage were explained and members confirmed the modified design proposal. The specification document, which was individually prepared, was confirmed by all members in the meeting. In this meeting, the shared applications were used frequently for integrating and confirming the final outcome.

3.6.2.3 Summary of observation

The observations described in the previous sections represent an example of a typical team design project and its processes. The overall phases do not seem to be significantly different from a design process accomplished by a designer working alone. However, in the team design project, social interactions between members were an additional aspect to be considered whilst making decisions. It must also be considered that group characteristics can vary according to the composition of the team.

Based upon observation and previous design process models, it can be concluded that the team design process includes several significant phases. It starts by identifying the design problem, building common approaches and unifying understanding of the design problem. Designers are then involved in generating alternatives and their critical evaluation and refinement. Next, production of design proposals and detail refinement are followed during the production of final design proposals.

Current collaborative technologies allow distributed designers to accomplish a team design project without having face-to-face meetings. However, in order to perform efficient and practical team design work, many technical issues need to be addressed. Such issues include the way to co-ordinate concurrent activities, the speed and quality of video and audio conferencing tools, the efficient configuration of collaborative tools and the support of multi-user interface in computer based tools.

3.7 Discussion

3.7.1 Comparison of the two projects

One clear difference between the two projects was the amount of information exchanged between team members throughout the project. The first team engaged in verbal and visual information exchange much more than the second and spent a longer time at each meeting. The number of ideas and alternatives discussed during the face-to-face meetings were much larger than in the electronic meetings. Participants had difficulties sharing proposed concepts in the electronic meetings.

The first team showed distinct social aspects of team project management, while in the second team they were less apparent. For example, one member in the team quickly led the team and played a leader role, making decisions and managing the project schedules. On the other hand, such role playing and social interactions were not observed in the team using electronic meetings.

The atmosphere of the face-to-face meetings became relaxed and informal during the later phases of the project. It appeared that there was a common sense of the project and awareness of the project's progress. Informal discussion by designers in the second team was only concerned with the project.

In face to face meetings all kinds of media including sketches and listings were used both on paper and through computer rendering and electronic documents. Whereas the designers in the electronic meetings were limited to digital media.

Workspace activities during the meeting looked different for the two teams. Individual sketches were instantly shared by all members in the face-to-face meetings. All the members were expected to focus on the same issues. An individual workspace was not easily maintained because most of the activities at the discussion table were shared with others. On the other hand, in the electronic meetings, sketches or drawings were not shared until captured by the camera or explicitly described. The computer screen and drawing space were dedicated to the individual workspace.

In terms of the design process, both teams reached their main decision at the end of the second phase. Once the decision was made, any change to the design direction required acceptance from all members. Therefore, there was less flexibility to decision making and approaches to new ideas. At the later phase of the design project, both teams divided tasks concerned with the production of the final outcome, such as creating dimensioned drawings, 3D renderings and writing technical specifications. Table 3.6 summarises the comparison.

Table 3.6 Comparison between the project with face-to-face meetings and that with electronic meetings

Criteria	**Team project with face-to-face meetings**	**Team project with computer mediated meetings**
Amount of information exchanged	Dynamic and large amount of information	Small amount of information and a shorter meeting time.
Social aspect of team project	Apparent	Less apparent
Meeting atmosphere	Relaxed and informal	Formal and rigid
Media used	Traditional mixed media	Electronic media
Workspace activities	Mainly shared workspace	Combination of shared and individual workspace
Team design process	Identification, decision, detail refinement and production of proposal	Identification, decision, detail refinement and production of proposal

3.7.2 Collaborative tasks

Several types of collaborative tasks were observed in the face-to-face meetings and the significance of these varied as the project progressed. Four kinds of collaborative tasks were apparent: the conference task, the co-working task, the information exchange task, and the management task. Conference tasks took place when ideas were negotiated and exchanged. This was particularly important in the early decision making phases, when many issues were discussed. The co-working task was observed when an outcome was produced collaboratively. This became a major activity in the later phase when the team was involved in the shared production of the design outcome. The information sharing task was an integrated part of the project. Teams frequently shared what they had prepared during individual sessions when it was necessary to share tasks more effectively. Managing activities were also observed occasionally in all phases.

3.7.3 Tools to support each phase of the team design process

Based upon the observation of collaborative tasks, it is possible to suggest *four types* of collaborative technologies to support team design activities in distributed situations: resource sharing tools, communication tools, process management tools and shared design workspace tools to support synchronous and asynchronous co-working.

Throughout the design project, resource sharing tools can keep a record of design concepts and allow everyone in the group to readily share design information. Functions of the shared repository tools might include storing the evolution of design concepts and easy access to the information archives. It must also be considered that the shared repository tools need to keep both digital and traditional media, since designers frequently use sketches and listings for initial concept development.

Tools for effective communication allow designers to share proposed ideas among team members more easily. In particular, these tools can be effectively used in the conceptual design phase of team design projects, as individual ideas need to be easily represented to all group members. Collaborative systems combining features of workspace sharing and inter personal communication can be used to support effective communication.

Shared applications allow multiple designers to refine various concepts in parallel. As shown in the experiment, shared window applications have many limitations in a real-time multi-user situation. In order to manage a true multi-user environment, these tools should address issues such as awareness, access and concurrency control and group management. More sophisticated co-working tools, such as co-authoring, co-building and co-drawing tools, might be necessary in the detail design phase of the team design project. Consideration of co-working tools is important, since much design refinement takes place during the co-working phase. Real-time co-working tools could be a new domain for computer based design tools.

Project management tools should be a part of the collaborative design environment, as they are essential for a systematic approach to the team project. Those tools are necessary to allow designers to systematically allocate work to all members during project planning. Frequent informal interactions between group members can help users to make sure that the updated solutions are satisfactory to all members. Checking and confirmation tools might be used to see if there are any features missed during the development process. Once these tools are customised for team design activities, they should be smoothly integrated to build an efficient collaborative design environment.

3.8 Conclusion to the preliminary chapter

A primary objective of the preliminary study was to clarify research problems for the investigation of collaborative design environments, in particular issues to support real-time shard 3D design workspace activities. It was not intended to carry out an in-depth analysis of team design activities in face-to-face and in computer mediated design meetings. Instead, it suggested some insights on computer support for collaborative design using currently available technologies. The observation of design teams carrying out this experimental project showed that a set of existing collaborative technologies could support distributed designers completing a design project. Designers experienced difficulties while using available collaborative technologies in the computer-mediated environment because of limitations such as the communication speed and the limited screen resource. Nevertheless, designers could adopt tools in the computer-mediated environment and successfully accomplish the project. The comparison of two conditions also shows the feasibility of existing collaborative tools.

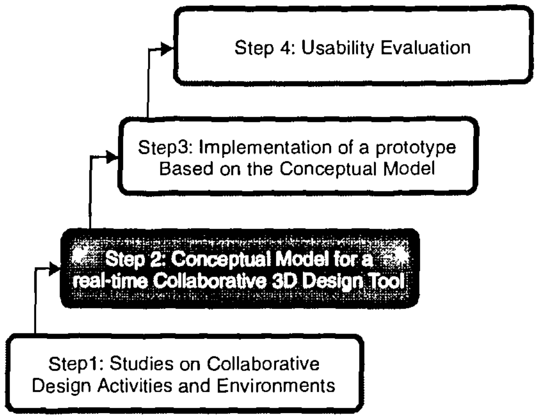
Much of what was observed in the experimental project was predictable for design teams. However, the result clarified the nature of team design activities and processes, and suggested that the characteristics and roles of the meetings changed as the project progressed. The research technique used to compare the two sessions can be used for examining the impact of other new tools in a collaborative design environment. Four types of technology based upon collaborative tasks were identified and suggested as the components of a collaborative design environment in a co-located or distributed situation. These include tools for resource archiving and sharing, communication, process management and co-working.

Most of the currently available collaborative technologies have been developed for general meeting environments and do not provide specific design tools, such as sophisticated 2D and 3D graphics applications. A combination of tools, not originally developed to support design activities, could not address specific user interface issues in the collaborative design environments. It is therefore necessary to investigate a novel system originally designed to support fundamental aspects of the collaborative design environment.

As one of the novel systems for collaborative design environments, the following chapters focus on a real-time collaborative 3D CAD system. Collaboration-aware 3D CAD systems were not provided in the preliminary experiment but are apparently essential for a team design project in various design domains, such as product and architectural design. There are many challenges for a multi-user 3D CAD system to allow multiple designers to collaborate simultaneously during 3D concept generation and detail refinement. Issues specifically related to supporting real-time multi-user 3D design tasks in a collaborative design environment are identified and a framework to address some of the issues is also suggested and evaluated. The lessons of preliminary studies can be used to consider the ways of integrating the novel system into a collaborative design environment.

Chapter 4

A Conceptual Framework for a Real-time Collaborative 3D CAD System



4.1 Introduction

Having taken a broad look at collaborative design environments through the preliminary investigation, this chapter focuses on a conceptual framework for a real-time collaborative 3D CAD system. The properties of shared workspaces, used primarily for design processes, are examined and compared with general-purpose shared workspaces. The usage of 3D design tools is analysed in order to illustrate 3D design activities and the limitations of existing single-user 3D CAD applications in a collaborative situation. The requirements of a new real-time collaborative 3D design tool are specified. Then the conceptual model, 'the Shared Stage', is proposed as a means of providing real-time collaborative 3D design features. In order to develop the conceptual model into a prototype system, various environments and tools are reviewed for the prototype implementation. The reasoning behind the selection of the prototype environment and the details of the environment are also presented.

4.2 Shared workspace in design

4.2.1 Physical and virtual shared design workspaces

A shared workspace is a physical space where people can undertake some joint activity (Gutwin, 1997). For design processes, the *physical shared workspace* could be a meeting room, a drawing table and a modelling workshop. In these workplaces, designers occupy their individual workspace as well as common workspaces. Sometimes the physical division of the common and individual workspaces is not clear. This is typical of a team design workplace where there are frequent and dynamic interactions within the group.

The use of computers in the design process has provided another kind of workspace. This new workspace is located within computer based design tools and can be referred to as the *virtual design workspace*. Throughout the design project a large amount of information is saved and processed in electronic forms. Hence, the virtual design workspace that involves the use of computer-based tools plays an important role in the design process. These virtual design workspaces have also substituted some physical design workspaces such as drawing tables and modelling benches. As the use of virtual design workspaces increases, designers spend more time in front of computer screens. Although virtual design workspaces are becoming more prevalent, their collaborative use has so far been limited.

4.2.2 Properties of shared design workspaces

The term 'Shared Workspace' has been widely used in the fields of CSCW and HCI. Some differences can be found between the characteristics of shared workspaces for design activities and conventional shared workspaces.

Gutwin (1997) listed several properties of a typical shared workspace. He suggested that interactivity and awareness are key properties of the shared workspaces. Shared workspaces provide an environment for interaction, thus giving users something of which to be aware. Interaction and awareness provide perceptual availability, enabling them to observe others as they move about the space and work on artefacts. Spatial organisation is another property of the shared workspace. In joint activities location and spatial relationships are used within the shared workspace in a meaningful way. The shared workspace also has the property of a bounded environment constraining interpretation and allowing users to map information such as movement or sound onto meaningful events in the activity. These properties characterise general-purpose shared workspaces that may be seen in many office environments.

Such workspaces usually consist of textual information and the output of the activities taking place within the shared workspace is essentially two dimensional documentation. Many collaborative activities in the office environment are processed in sequential ways and a certain workflow can be observed. Therefore, asynchronous interaction by exchanging documentation is commonly used and can be an efficient method of collaboration. BSCW (Basic Support for Co-operative Work) (Bentley et al., 1996; Koch and Appelt, 1998) and Lotus Notes (Lotus, 2000) are example applications which support sharing, primarily document based information, within an organisation. These applications facilitate document uploading and downloading with group management features. The role of a shared information repository is emphasised rather than a real-time interaction between those involved.

For design workspaces, however, there are some specific characteristics that make it different from conventional workspaces. The first important feature of the design workspace is that the context is largely graphical or multi-media based. 2D sketches and 3D conceptual models are important parts of the design workspace. Such graphical and multi-media information should be integrated and exchanged freely in the shared workspace. Where difficulties of communicating design rationale exist, more frequent interaction between collaborating designers is required through the shared workspace. The design workspace often involves the building and modifying of 3D models. As 3D models are visualised by means of drawing, sketches or physical models, designers require tools that aid discussion regarding changes to these 3D models. With the increased use of advanced CAD systems, designers often work together in front of computer screens whilst undertaking 3D design tasks. Another aspect of the design workspace is that design workspace activities require more continuous and dynamic interactions. It is common for design ideas to constantly evolve throughout the design process. Thus, there are greater needs for real-time collaboration in the shared design workspace than in other workspaces. Table 4.1 summarises the comparison of the properties between shared design workspaces and general-purpose shared workspaces.

Table 4.1 Comparison between general shared design workspaces and shared design workspaces

	General shared workspace	Shared design workspace
Context	Documentation	2D and 3D artefacts
Information	Primarily textual information	Graphical and Multi-media information
Type of collaboration	Asynchronous and sequential	Synchronous and interactive
Process	Structured Workflow based	Unstructured, iterative Dynamic

4.3 Computer based 3D design activities

In a physical design workspace, 3D design models can be constructed by combining pieces of wood or sculpting a form using clay. In virtual design workspaces, designers work on 3D models using particular construction methods provided by the available 3D CAD systems. For example, using a surface-based 3D CAD tool, a designer can accomplish various tasks, such as creating surface geometry, applying shading parameters and textures, lighting the scene and camera positioning. The model is rendered and may be combined with photo images of the environment.

In order to create surface geometry, a designer first constructs a basic form from profile curves and simple primitive objects, such as cubes, spheres, cylinders and cones. By combining, stretching or cutting through the simple primitive objects, it is possible to build moderately complex forms. More sophisticated surface geometry can be generated by revolving, skinning or extruding the 3D profile curves. Surface editing tools such as trimming or intersecting tools may also be used. Objects are scaled, rotated and moved using various transformation techniques to create the desired shape. The constructed surfaces are combined or grouped to become a complete 3D object assembly. At this stage, 3D geometry can be exported to other CAD systems for further analysis, virtual simulation, rapid prototyping, or manufacturing. Much of the above also applies to solid modelling CAD applications.

Once objects are created and transformed to a desired shape, a designer applies shading parameters and textures to produce a realistic computer rendering. The designer should decide on a type of shading according to the purpose of the rendering. There are several shading models, and each shading model has its own shading parameters. Examples of the basic shading parameters are colour, specula, incandescence, reflection and diffuse. Textures are applied to these parameters to create more sophisticated visual effects. Basic types of textures involve surface textures, environment textures, and solid textures. Surface or environment texture mapping places a two dimensional texture in the parameter space of a surface, while solid texture mapping is a three dimensional texture which makes a surface appear as if it is carved out of a solid block. The designer decides shading parameters and textures by repetitive experimentation with constructed objects.

Once objects are modelled and shading parameters are applied, the rendering process follows. To render a scene, it is necessary to set the lights and aim the camera. The designer should decide light types and locations and rendering parameters. Typical types of lights include ambient, directional, point, linear and area lights. Each light has parameters that affect how it illuminates a scene. Examples of light parameters include intensity, shadows and colour. The designer also needs to aim the camera at the objects to get the best shot of the scene. Typical camera movement includes dolly, zoom, tumble, and pan. For the final rendering process, the designer decides which rendering type is required for the final representation. Rendering types can be hidden-line, ray cast and ray trace. The final rendered images may be combined with the other photo images of physical environments in order to examine how the model matches with the environment in which it is used or located. Although designers follow a certain workflow as required by 3D modelling applications, these activities are not necessarily sequential because designers perform iterative refinements to build a model that suits their design goals.

4.4 Issues of a real-time collaborative 3D CAD system

There are many situations where real-time collaboration between multiple designers is required for the computer based 3D design activities described above. For example, constructing a simple 3D model may be done effectively by a single designer using current 3D design tools. However, if the form is complex and consists of many interconnected parts, it takes a long time for the single designer working alone. Sometimes 3D models must be constructed within a very short time period to explore specific features of a design idea. Real-time collaboration in discussing and modifying the shared 3D models can enhance the efficiency of the team design process.

4.4.1 General issues

In the development of a real-time collaborative 3D CAD system, a starting point is to consider general issues for real-time application sharing. These issues can be classified into technical and user interface issues. Technical issues are concerned with session control, floor control, concurrency control and appropriate system architecture for a multi-user environment. The user interface issues include the need for a better sense of tele-presence in the shared workspace, the ability to have a seamless incorporation of shared work on the computer and real world and consideration of the flexibility of group work.

4.4.1.1 Session control

A session is a situation where a group of people are in a shared workspace together at a given time (Brink, 1998). Session control is concerned with finding out what collaborative sessions are available, determining who can join and leave the session, and when and how they may do that. For some real-time collaboration, anyone may be allowed to join, whereas for others only a selected group can participate or newcomers should be invited by existing session users. Many other considerations can be given to session control mechanisms. Session control may allow people to join and leave at any time, but intrusive situations where users are able to invade privacy or impose a session on others should be avoided. Sometimes more spontaneous creation of collaborative session is desired for session control (Kraut et al., 1990), while other situations require a central facilitator to handle registration (Nunamaker et al., 1991).

4.4.1.2 Floor control

Floor control is concerned with the decision of what kind of access each person has to the shared contents during a real-time collaborative session. For example, when using a shared whiteboard, a floor control mechanism should decide whether only one person can access the shared drawing space at any time or a moderator can control the access. Time limit can also be applied to manage the floor control.

Simultaneous access by everyone involved in a collaborative session is often preferred as it gives fluid interaction, but it might cause problems with a large number of people. An abuse by a single person may interrupt the entire workspace activity. Some kind of mediated access can be applied to prevent mistakes, unauthorised access and people making conflicting changes. A combination of these floor control mechanisms would also be possible. Most importantly, smooth floor control mechanisms should be applied to prevent interference with the work of others or unexpected interruption.

4.4.1.3 Concurrency control

Concurrency control is often needed to mediate access to the object. It is required to ensure that a state of replicated shared workspaces remains consistent even when users attempt to modify the objects simultaneously with a sequence of operations (Prakash, 1999). For example, two users may attempt to modify an object in a replicated shared workspace via two operations. If each operation is executed locally first and then broadcast for execution at other sites, the operations would be applied in different orders at the remote sites. This potentially leads to inconsistent states and results in an undesirable situation in general.

There are two broad classes of concurrency control techniques: pessimistic and optimistic. Pessimistic techniques ensure that inconsistencies among copies do not arise by requiring that any update operations acquire appropriate locks to prevent conflict updates from occurring. Optimistic techniques do not prevent inconsistencies from occurring, but use mechanisms to detect and correct inconsistencies if they occur. The pessimistic technique can be achieved through a simple locking, ordered broadcasting of protocols (Ellis et al., 1991; Birman and Joseph, 1987; Chang and Maxemchuck, 1984) and using a centralised data architecture. The optimistic techniques are more complicated and concerned with undo/redo mechanisms in a multi-user environment (Karsenty and Beaudouin-Lafon, 1993). However it is used often with the goal of reducing interactive response times.

4.4.1.4 System architecture

It is also important to consider the merits and drawbacks of three possible run-time architectures of real-time collaborative applications: centralised, replicated and hybrid architectures (Greenberg and Roseman, 1999). Centralised architectures use a single application program, residing on one central server machine, to control all input and output to the distributed participants. Client processes are responsible only for passing requests to the central program and displaying any output sent to it from the central program. Synchronisation and concurrency control is simple, but the central server could become a network or performance bottleneck. Replicated architectures, on the other hand, execute a copy of the program at every site. Thus each replica must co-ordinate explicitly both local and remote actions. Performance and response time can be improved, but special consideration should be given to the ways of synchronising the state of each copy. Hybrid architectures contain both centralised and replicated components. It might be possible to maintain state information centrally while allowing users at the replicated sites to decide what the view should look like and to update the display accordingly (Patterson et al., 1996).

4.4.1.5 Awareness support

Many group work situations benefit from implicit communication, such as indirect gestures, information about people's environment, or biographical information about people in a collaborative session. This information helps people to establish common ground, co-ordinate their activities, and helps avoid 'surprises'. Consideration should be given to provide this information without obstructing privacy or work progress.

4.4.2 Issues of collaborative 3D CAD

In addition to the general issues described above, there are some issues that are particularly important for real-time collaborative 3D CAD. These issues include

- Sharing intermediate processes
- Dynamic and interactive exchange of intermediate 3D data
- Support of collaboration for integration activities
- Incorporation of individual and shared workspaces
- Maximising acceptability of new tools by consistent user interface
- Awareness support in the shared 3D workspace

4.4.2.1 Sharing intermediate processes

Tang (1991) emphasised that sharing intermediate processes facilitates collaborative design activities. When working alone, a designer can make changes or try alternative ideas freely at any stage of the design process. However, in a team design environment, all the members of a group need to have a shared understanding of the progress and agree over changes. When two or more designers work together on a shared 3D design task, if they are aware of the activities of other participants they can easily can help each other to solve particular technical or design problems. By sharing information about the process, they can automatically see problems and be aware of the methods members have used to try to solve them.

4.4.2.2 Dynamic and interactive exchange of intermediate 3D data

Real-time collaborative 3D CAD systems should allow designers to freely exchange the intermediate outcome without any miscommunication during a collaborative process. Because of complexity in the design outcome, semantic and structural information may need to be shared to aid participants' understanding. A common way to share intermediate outcome using existing tools is to exchange data files. Files are multiply saved and distributed via file exchange tools. Confusion may arise with unstructured and asynchronous access to the distributed data files. Dynamic and interactive mechanisms of data distribution can help designers to track the evolution of the design model during a real-time collaborative session.

4.4.2.3 Support of collaboration for integration activities

Different parts of complex shape can be modelled by different team members using existing single user oriented 3D modelling tools. However, the integration of the parts is mostly accomplished by one person using existing tools because there is no collaboration support in the integration phase. Inconsistency between different parts constructed by different designers should be carefully considered. Modification of inconsistent parts for effective integration may take even longer than constructing a new set of parts. Collaborative 3D design tools should therefore allow designers to work together efficiently in such integration tasks having the efficient 3D interface.

4.4.2.4 Consistent user interface

The way to provide maximum usability of new real-time collaborative 3D features to current 3D CAD users is also important in the development of a new tool. Because of the complexity of the interface of 3D CAD systems, it is necessary to learn to use the tools available, in order to operate 3D design tools effectively. Once designers are familiar with 3D design tools, they can readily transfer their skills to a new tool having a similar interface. Therefore, to make a new interface to the real-time collaborative 3D design tool highly acceptable to designers, consistency in the interface of current 3D CAD systems should be taken into account.

4.4.2.5 Incorporation of individual and shared workspace

Individual 3D workspaces play an important role in real-time collaborative 3D design. It is usual for multiple designers working in a large scale 3D CAD project to subdivide parts of a complex model and then integrate them into a whole assembly. During this process, designers need to switch their focus frequently between individual and shared 3D workspaces. They need to maintain their own individual 3D workspaces at the same time dynamically accessing the shared 3D workspaces in order to check the integrated status and to exchange information about progress.

4.4.2.6 Awareness support in shared 3D workspaces

It is also important to provide appropriate awareness information in a collaborative 3D environment. Designers can work more effectively if they know where collaborating designers are working and observing, what they have created or modified and what they are doing in shared 3D workspace (Gutwin et al., 1996; Gutwin and Greenberg, 1998). In 2D collaborative environments, a radar view of the shared workspace or a multi-user scroll bar can be used to support that information. However, as such features are inappropriate in a 3D shared environment, it is necessary to find ways of providing appropriate 3D workspace awareness information with minimum cognitive loads for the designers.

4.5 Shared Stage: A conceptual model for real-time collaborative 3D CAD

In order to address the issues mentioned above, a shared 3D workspace for a real-time collaborative 3D CAD, called 'Shared Stage', is proposed. The Shared Stage is a collaborative 3D workspace to be added to a host 3D CAD system. As the user interface is based on the host 3D CAD application, users can rapidly adapt to the new real-time collaborative features.

A number of potential features can be incorporated into the Shared Stage to support real-time collaborative 3D design activities. 3D objects created by multiple users can be integrated automatically into the Shared Stage in real-time. Thus collaborating designers can instantly examine how their work is related to other work. Synchronised 3D representation and multi-user cursors may allow designers to discuss their work during a 3D design task. Objects in the Shared Stage may be displayed in a different colour to represent multi-user information, such as ownership. Various awareness information, such as who they are working with, what they are working on and looking at, what they have created and edited, can be provided in the Shared Stage. The Shared Stage can also be used as a place for a dynamic exchange or referencing of individually created 3D objects.

Figure 4.1 shows an example workspace environment of a real-time collaborative 3D CAD session with the Shared Stage. In this figure, each user has access to their own individual 3D CAD workspace that is linked to the Shared Stage. When designer A creates an object such as a cube in their own workspace, the object instantly appears in the Shared Stage. Modification or deletion of the cube is also updated instantly. When user B creates a sphere object, the same update sequence takes place. Both users can observe not only their own objects but also objects created by their partners at the same time. As they may observe the status of the Shared Stage at all times, it is possible for them to be aware of their partners' procedures. They can discuss the model and ways to construct different aspects of it as they share the same representation and have a means of communication in the Shared Stage. Users can transfer objects one another for the purpose of examination or modelling, or may undertake further modifications instantly through the Shared Stage.

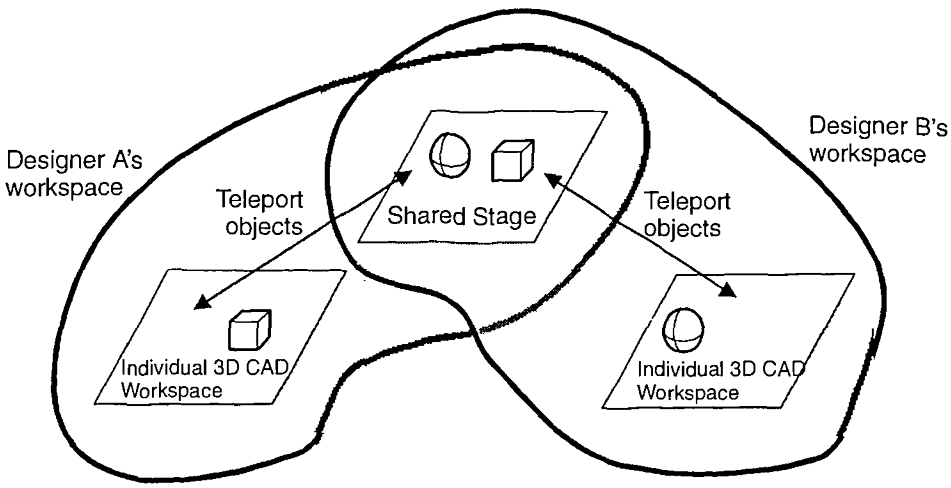


Figure 4.1 The workspace environment of a real-time collaborative 3D design session with the Shared Stage

From the user interface perspective, the use of the Shared Stage can be compared to a situation where a user has two types of special 'viewers' whilst working in a 3D workspace (Figure 4.2). One type of viewer (for individual work) allows visualisation of the individual workspace and the other type (for collaboration) allows visualisation of the Shared Stage that represents all models being created by a group. In each case, designers use the same tools. When there is a need of collaboration in the workspace, users simply change 'viewers' from the individual 3D workspace to the shared 3D workspace.

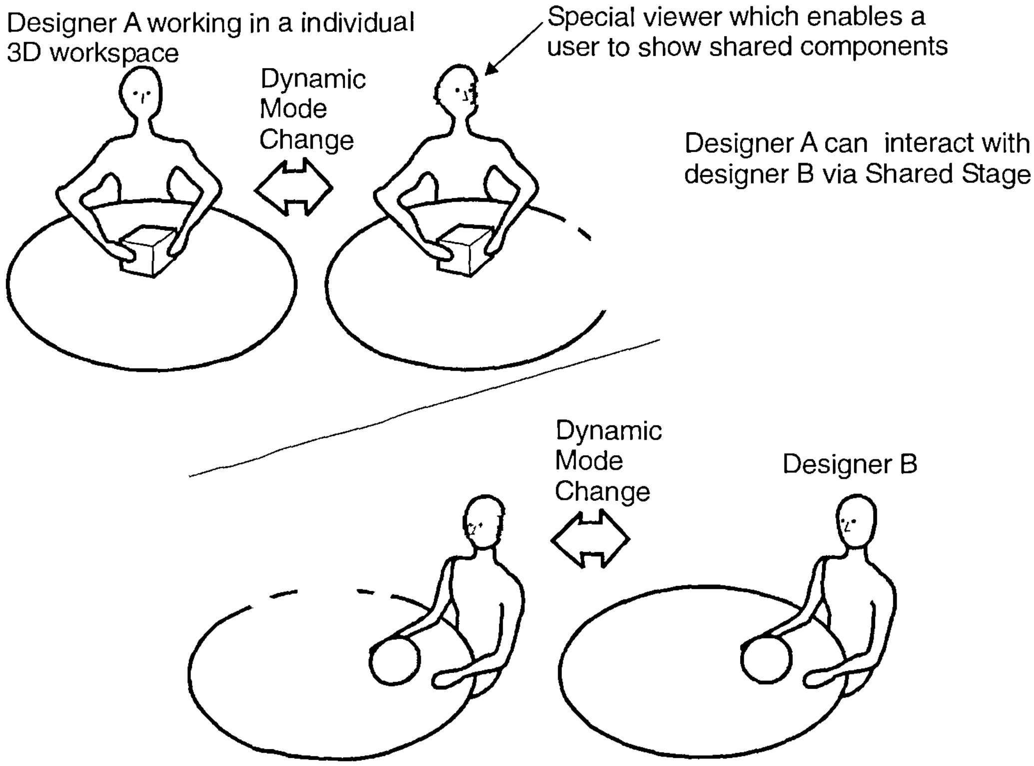
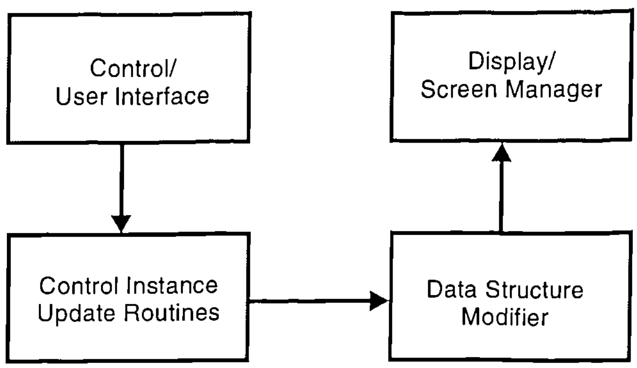


Figure 4.2 Shared Stage from a user interface perspective

The Shared Stage module can also be compared with a specific plug-in application for real-time collaboration. As plug-in applications are designed to extend the capabilities of host programs for specific added functionality, the Shared Stage module supports additional features for real-time collaboration. Users can comfortably accept the collaboration aware system transformed from existing familiar design tools. By maintaining the same user interfacc as the host design tool and extending it with a collaborative extension, design tools can be used seamlessly in both individual and shared 3D workspaces.

4.6 Software structure of a real-time collaborative application

Figure 4.3 shows a typical structure for a single-user oriented interactive software application, such as a text editor or drawing system. There are four general components in the application: user interface input, control instance update routine, data structure modifier and display manager (Parakash and Knister, 1992). In a 3D design system, for example, the user interface input part of the application manages all the input events from the user. When an input is received, the control instances are updated according to the type of input event. A change to the control instances is translated to the data structure modifier which updates the 3D data in a data buffer. The changed data triggers the display manager to show an up-to-date representation. The display manager usually represents data in a 3D format. In fact, users only interact with software through the user-interface component and the display manager.



Fi ure 4.3 Conventional software model of single user (2D/3D design) applicat'ons

The use of screen sharing is one simple way of supporting a real-time collaboration feature. It allows users to share user interface and display components with other participants. The 'Timbuktu' shared application environment, used in the preliminary investigation (Chapter 3), is an example of a screen sharing application. One disadvantage of screen sharing systems is the reduced flexibility of individual work because a users' entire screen is taken over by those of other users. A system like

XShare allows a single-user application to send the same output to multiple consoles (Tee, 1993). In this kind of system, usually based on X windows, a pseudo-server intercepts low-level protocol messages between the server and desired clients and multicasts the messages to X servers that are controlling other consoles. In this system also, the entire interface of an application must be shared. Since the application is not collaboration-aware, simultaneous input is normally managed by some floor-control mechanism. Demands on the network bandwidth are usually heavy because all display updates are broadcasted.

An alternative technique for transforming a single user oriented application into a collaboration-aware one is shown in Figure 4.4 (Parakash and Shim, 1994). An input broadcasting facility, between the user interface input component and the control instance update routine, intercepts user actions from the user interface component and broadcasts them to the remote broadcasting facilities. The remote broadcasting facility updates received user actions in the same way that it would happen locally. This approach normally requires a distinct division of roles in shared mode in order to manage floor-control. A user becomes an application master who controls the application or an observer who can only see changes in the application. The floor-control status may be exchanged between users. This prevents multiple users in the session controlling the workspace at the same time. Furthermore, an efficient concurrency control mechanism is of great importance since control broadcasting should be received in the same order for each participating user. However, the modification of a program structure in this way requires a great deal of effort, because the broadcasting facility is rigidly connected to other routines of the program. In addition, there are limitations imposed by the restricted access to the software code. Although many commercial applications now provide some access to the application structure, with customised development tools such as an API environment, partial development facilities may not be enough to modify the structure of the application.

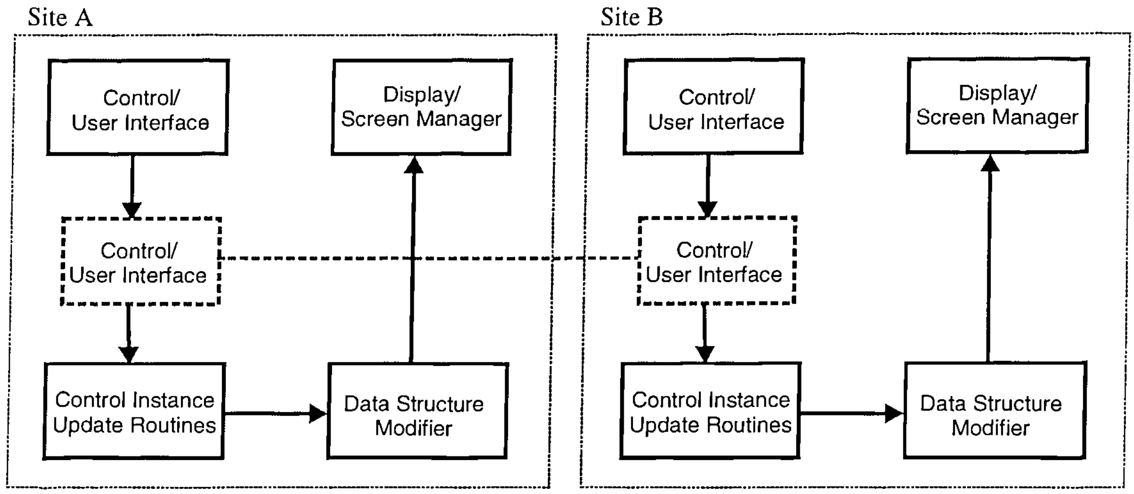


Figure 4.4 Transforming a single user application to a real-time distributed collaborative application (Parakash and Shim, 1994)

Figure 4.5 shows a conceptual software structure of a real-time collaborative 3D design system with the Shared Stage. The grey areas show the software component of the Shared Stage added to a host 3D design application. The Shared Stage may include four additional components: database replication and shared status co-ordination routines, replicated shared workspace database, user interface and display manager of the shared stage. When the data structure is changed within the host 3D design application, the database change is translated into the replication component of the Shared Stage module as well as the local display manager. The data replication component is connected to the central session manager and the shared database manager, which updates the global status of the database and distributes the co-ordination message to all connected shared status co-ordination routines. The central session manager manages the coherency of shared data as well as the control of concurrency. The shared status co-ordination routine creates a simplified replica of the whole content of the shared workspace database, according to the messages received from the central shared database manager. The replicated shared database forces the shared stage display manager to display the contents of the shared workspace. This conceptual software structure is the foundation of our prototype development to be described in the next chapter (Chapter 5).

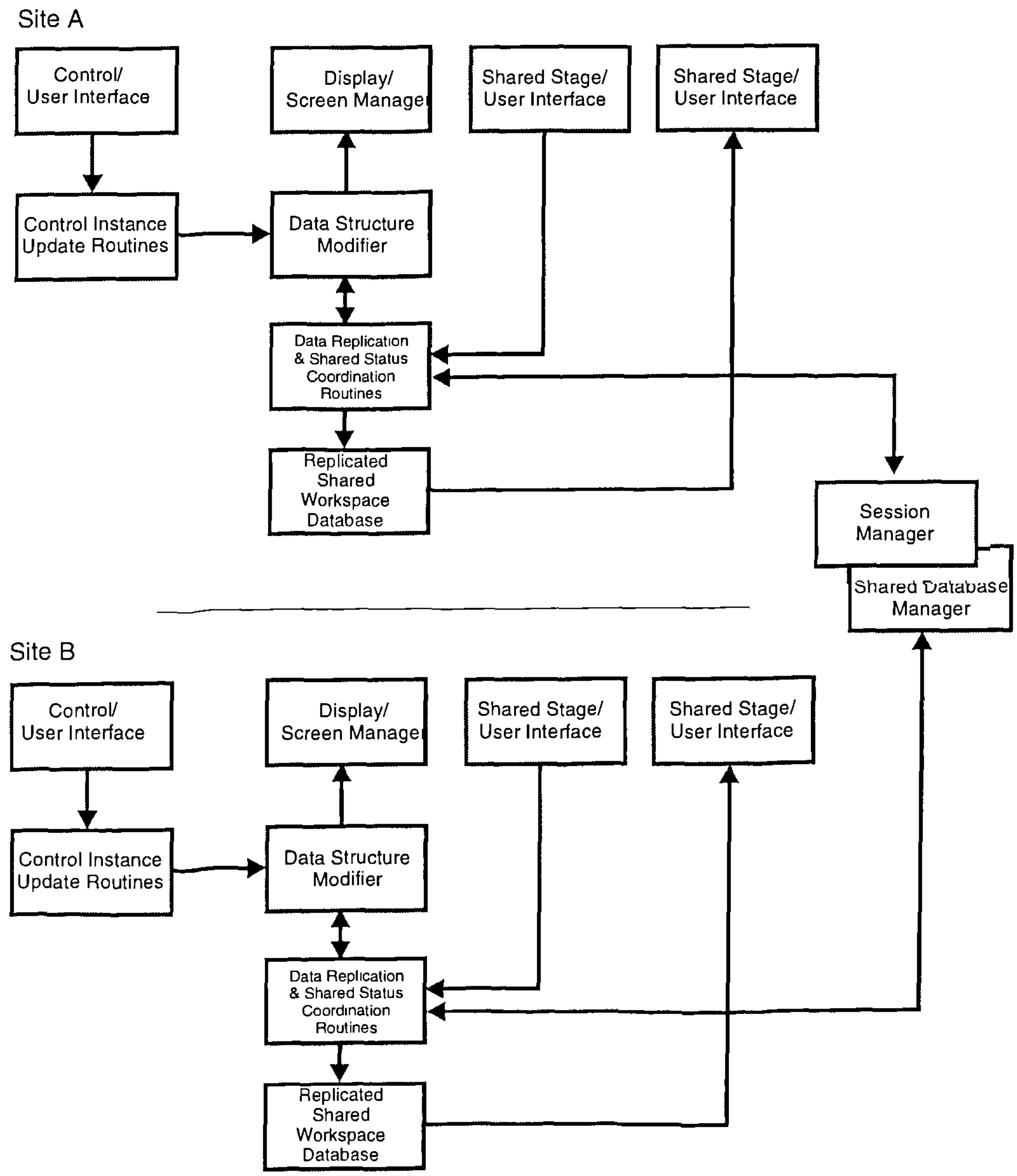


Figure 4.5 Collaboration-aware application with the Shared Stage extension

4.7 Software development environments for real-time collaborative applications

In order to implement the idea of the Shared Stage, many features had to be provided in the software development environment. The software development environment should provide effective programming capabilities. Coding should be effective and easy to allow the prototype to be developed and evaluated in a short time scale. Also, functions for real-time collaborative features should be easily integrated into the environment. Finally, the development environment should minimise the consideration of basic graphics functions to be implemented in the prototype system. In order to choose an appropriate software development environment for the implementation of the Shared Stage, software toolkits for real-time groupware and 3D graphics have been reviewed.

4.7.1 Toolkits for real-time collaborative applications

Software toolkits aim to provide software developers with an effective software development environment using various pre-defined functions or libraries. In the area of real-time collaborative applications, several toolkits have been developed. These include Harbanero, GroupKit, Rendezvous, Clock and Clockworks, and COAST.

4.7.1.1 Harbanero

Harbanero is a Java based API which is designed to develop general applications, such as shared text and graphic editors (Chabert et al., 1997). Its underlying framework is based on state and event synchronisation. Since it is Java based, application usage is cross-platform. Harbanero provides state and event synchronisation by replicating applications across clients and sharing all state changes in those clients. Serialisation of events is used for sharing. Floor control is adopted by the object arbiter, which can provide student-teacher or turn-taking methods. Access control is provided by locks, which restrict access to application resources. Session management facilities allow users to create, join, leave and browse sessions. Notification and awareness of others is also provided as a default. Example applications which use Harbanero include Co-ordination (a voting tool), Communication (Text and audio chats), Production (Shared Text editor, whiteboard), Weather visualizer and JavaGraph.

4.7.1.2 GroupKit

GroupKit is an extension to Tcl/Tk (Ousterhout, 1994) providing groupware features (Greenberg et al., 1995; Rossman and Greenberg, 1996). Using Tcl's built-in socket commands for its low level networking, GroupKit provides an application framework handling most details of building groupware, so users do not need to manage low-level groupware infrastructures.

GroupKit consists of a number of different processes. There is a central process called the registrar, a typical UNIX daemon that should already be running on the system. Each user runs a session manager, which connects up to the registrar. The session managers are used to create conferences which run as separate processes. When other users join conferences through their own session managers, GroupKit opens up network connections between the conference processes, so that every process in a conference has a connection to every other process in the conference.

There are two mechanisms for multi-user application programming in GroupKit: GroupKit Remote Procedure Calls (RPC) and GroupKit Shared Environments. GroupKit Remote Procedure Calls executes a Tcl command in the conference processes of different users. *gk_toAll* executes a Tcl command on all processes in the session, including the local user. *gk_toOthers* executes a Tcl command on all remote processes in the session, but not the local user. *gk_toUsernum* executes a Tcl command on a single conference process only, which may be one of the remote users or a local process. An alternative approach for multi-user programming in GroupKit is to use the features of shared environments. It is based on the idea of separating out underlying data from the view of that data and how it is manipulated. Environments in GroupKit store an arbitrary set of data, arranged in a tree structure, where users can associate a string value with any node in the tree. Environments can also generate events when they are changed, and these events can be received by other portions of the program. Parts of program handling user input can react to these changes by modifying the environment. The shared environment mechanism allows all users in a conference to share associated data. When any user makes a change to their copy of the environment, that change is automatically propagated to all other user's copies. So while a user's input event may change their own environment, generating an event to update their onscreen view, the change is sent to every other user's environment, generating the same event and updating their view. In fact, the shared environment paradigm is useful on larger applications because it helps to modularise the code.

GroupKit supports several concurrency control mechanisms: serialisation and locking. One of the RPC commands for serialisation is *gk_serialize* which works exactly like *gk_toAll*, except that the messages will be seen in the same order on all conference processes. Shared environments can be created as a serialised environment. Another method of concurrency control is to use a lock manager allowing a single conference process to hold a particular lock and perform a particular operation.

4.7.1.3 Rendezvous

The Rendezvous groupware toolkit is modelled on the idea of maintaining a single abstract data model that is shared by everyone (Hill, 1992; Hill et al., 1994). Rendezvous provides support for managing a multi-user session, for performing fundamental input and output activities, and for controlling the extent of sharing of information and control. Rendezvous focuses on two major requirements: providing flexible control over the dimensions of sharing and robust session management. Three dimensions of sharing have been identified: the sharing of underlying objects, sharing of the presentation and the sharing of the input authorisations or access. It was claimed that a system architecture must provide flexible control over these dimensions of sharing. In addition, a robust session management is required to allow users to join and leave as necessary without halting the session. Example applications using the Rendezvous architecture include a multi-user card game and a multi-user training simulation. The drawback of Rendezvous is that it is slow because the data model and propagation of constraints are centralised. It has only been developed for UNIX operating systems and X windows.

4.7.1.4. Clock and ClockWorks

ClockWorks is a tool for programming software architectures, so is somewhat simpler than other general visual programming languages (Graham et al., 1996). The Clock language is intended to support the prototyping of graphical user interfaces, concentrating particularly on the development of multi-user interfaces and groupware. Clock consists of a visual architecture language, used to specify the class, visibility and compositional structure of Clock programs. Clock is a declarative language intended for the programming of interactive systems. The components of Clock are programmed textually, in a functional language similar to Haskell (Hudak and Wadler, 1991).

Clock is based on an extended Model-View-Controller(MVC) paradigm of 'Smalltalk' (Krasner and Pope, 1988). In the MVC paradigm, the model represents the underlying state of the application. The view encodes how the user interface is to appear on the display. The controller specifies how inputs are to be handled. The key to MVC's power is the way in which these components communicate. User inputs are given to the controller, which then determines how these inputs are to be reflected in terms of modifications to the model. The model announces to the view then updates itself, resulting in a new display. This approach leads to a strong separation of concerns between the model, view and controller. The controller does not directly communicate with the view, only with the model. The view is responsible only for updating the display, and does not need to know the details of when they may arise.

4.7.1.5 COAST

The COAST framework employs a fully distributed and replicated architecture (Schuckmann et al., 1996). It intends to make the development of synchronous groupware easier in several ways. First, it is achieved by hiding the network from the application programmer. The system decides when each data object has to be transferred over which network channel, so that the application programmers do not need to handle transport of data or objects over the network explicitly. Secondly, the synchronous groupware development can be made easier by hiding the physical location of objects from the application programmer. The system decides at which physical locations to keep automatically synchronised replicas of each shared object. The physical location of shared data is transparent to the application programmer. Finally, concurrency is partially hidden. COAST includes a lightweight transaction system. By the use of transactions, the application programmer can comfortably handle global concurrency. The system optimistically ensures serialisability of all transactions. The COAST is written in VisualWorks Smalltalk and TCP/IP support is required. Because it uses the Smalltalk language, other toolkits that are required for the development of an interactive graphics application are difficult to integrate.

4.7.1.6 Comparison of toolkits

Most of these toolkits reviewed above aim to facilitate the development of general-purpose distributed *collaborative applications*, such as whiteboard, meeting support environments. The toolkits do not provide direct support for specific features, such as 3D visualisation. Furthermore, most of these toolkits are still in the research prototype stage, so are still evolving or development has stopped. In order to choose an appropriate software toolkit for the Shared Stage and a real-time collaborative 3D CAD system, each toolkit is examined in terms of such aspects as 3D graphics support capabilities, ways to choose various user interface methods, simple programming abstractions, base language and operating system compatibility. Table 4.2 shows the appropriateness of toolkits as the development environment for a collaborative 3D CAD system.

Table 4.2 Comparative review of real-time collaborative toolkits

Toolkit	Habanero	Groupkit	Rendezvous	Clock	COAST
Base Language	JAVA	Tcl	MEL: Common LISP	Clock Decrative Language	Smalltalk
3D Graphics Support:	JAVA 3D Graphics API	Tcl 3D Extension	NA	NA	NA
Means of implementing user Interface components	Java interface toolkit	Tk	NA	Clockwork	NA
Mechanism of Real-time support	State synchronisation	RPC MVC	Shared Single abstract data	MVC	Transparent Shared Data
Ease of use	Good	Very good	Bad	Good	Bad
Platform Independence	Very good	Good	Bad	Bad	Bad
Compatibility	Good	Reasonable	Bad	Bad	Bad

From the review of the development environments for real-time multi-user applications, GroupKit is considered as an appropriate tool for this study because it provides not only support for basic connectivity but also for high-level programming abstractions. For basic connectivity, it provides an infrastructure to support conference registration and subsequent communication between processes owned by conference participants. Since GroupKit was initially based on the Tcl language, it was decided to implement the prototype system with the Shared Stage using the Tcl script language. The Tcl language provides enough programmability to build the requisite complex application. It is also important that the environment should provide a high level programming interface for 3D CAD and graphics operations. The Tcl extension of the Visualisation Toolkit (Schroeder et al., 1996) was chosen as a 3D graphics toolkit because of its compatibility with GroupKit and comprehensive programming interface for 3D data visualisation and graphics. In the following sections, Tcl and the Visualisation toolkit extensions are briefly introduced and the way of customising them for a multi-user application is explained.

4.7.2 Tcl/Tk

Tcl is an interpretative language developed by John Ousterhout during the late 1980's (Ousterhout, 1994; Welch, 1995). It was designed to provide a flexible command language that could be easily integrated with a variety of applications. Tcl itself is written in the C programming language, and has a well-defined Application Programmers Interface (API) for integrating new functions. As a scripting language, Tcl is similar to other UNIX shell languages such as Bourne Shell, C Shell and Perl. In addition to essential programmability, such as variables, control flow and procedures, Tcl fills the role of an extension language used to configure and customise applications. By adding a Tcl interpreter, a programmer can structure an application as a set of primitive operations that can be composed by a script to best suit the needs of the programmer. Another strength of Tcl is that there are many extensions developed by the Tcl community which are freely available. The most notable extension is Tk, a toolkit for X windows.

4.7.3 The Visualisation Toolkit

The visualisation toolkit is an object-oriented 3D graphics toolkit which contains a number of software libraries and an interpreted Tcl/Tk environment designed for data visualisation and 3D computer graphics (Schroeder et al., 1996). There are several basic graphics objects used in the visualisation toolkit and a visualisation process which transforms raw data into a meaningful representation that graphics engines can render. Eight basic classes, which are most frequently used to render a scene, are *VtkRenderWindow, VtkRenderer, VtkRenderWindowInteractor, VtkLight, VtkCamera, VtkActor, VtkProperty and VtkMapper.*

A typical process used to create a rendered image of graphics objects using the Visualisation Toolkit involves several steps. The first step is to create a rendering window. A class called VtkRenderWindow ties the rendering process together and is responsible for managing a window on the display device. The instances of render are then created and linked to the render window. A renderer instance co-ordinates and maintains a list of the actors, lights, and an active camera in a particular scene. When render window and renderer instances are created, the instances of geometry, property and actor objects are created to represent actual graphics objects in the scene. The instances of vtkActor class combines object properties, geometric definition, and orientation in the world co-ordinate system. This is implemented behind the scenes by maintaining variables that refer to instances of vtkProperty, vtkMapper and vtkTransform. The actors are added to the renderer to be displayed in the scene. Camera and light objects can be added. If they are not specified, default camera and lights are created automatically when rendering is requested through methods of the renderer instance.

The visualisation process is defined as a process of converting data from its original form into graphics primitives (Schroeder et al., 1996). Visualisation deals with the issue of transformation and representation. Transformation is the process of converting data from its original form into graphics primitives, and eventually into computer images. Representation includes both the internal data structures used to depict the data, and the graphics primitives used to display the data. Transformations are processes in the functional model, while representations are the objects in the object model. The visualisation model in the Visualisation Toolkit is characterised with both functional models and object models. The example script below illustrates ways in which the Visualisation Toolkit objects can be used in actual graphics programming.

Example : Cylinder.tcl

```
# create a window to render into
vtkRenderWindow renWin;

# create a render and speficy it to the created window
vtkRenderer ren1;
renWin AddRenderer ren1;

# create sphere geometry using default parameters
vtkCylinderSource cylynder;
```

```
# map to graphics library
vtkPolyDataMapper map;
map SetInput [cylinder GetOutput];

# Create a property to be applied to an object
vtkProperty propCylinder;
propCylinder SetColor 1 0 0;# specify the color to red

# actor coordinates geometry, properties, transformation
vtkActor aCylinder;
aCylinder SetMapper map;
aCylinder SetProperty propSphere;

ren1 AddActor aCylinder;
ren1 SetBackground 1 1 1;# set background white

# Render an image; since no lights/cameras specified,
# they are created automatically
renWin Render;

# prevent the tk window from showing up then start the event loop
wm withdraw .;
```

The example shows a simple application to create and render a cylinder primitive within a graphics window. Figure 4.6 illustrates a pipeline used in the application. A vtkCylinderSource object(cylinder) is a data object that produces primitive cylinder data. It is specified as an input for a mapper object(map). The mapper object is a process object with inputs and outputs. It provides geometry information of a vtkActor object (aCylinder). A vtkProperty object (propCylinder) is created and a colour parameter is defined. It is also used as an input for the property of the actor object (aCylinder). These objects are connected to form a visualisation pipeline.

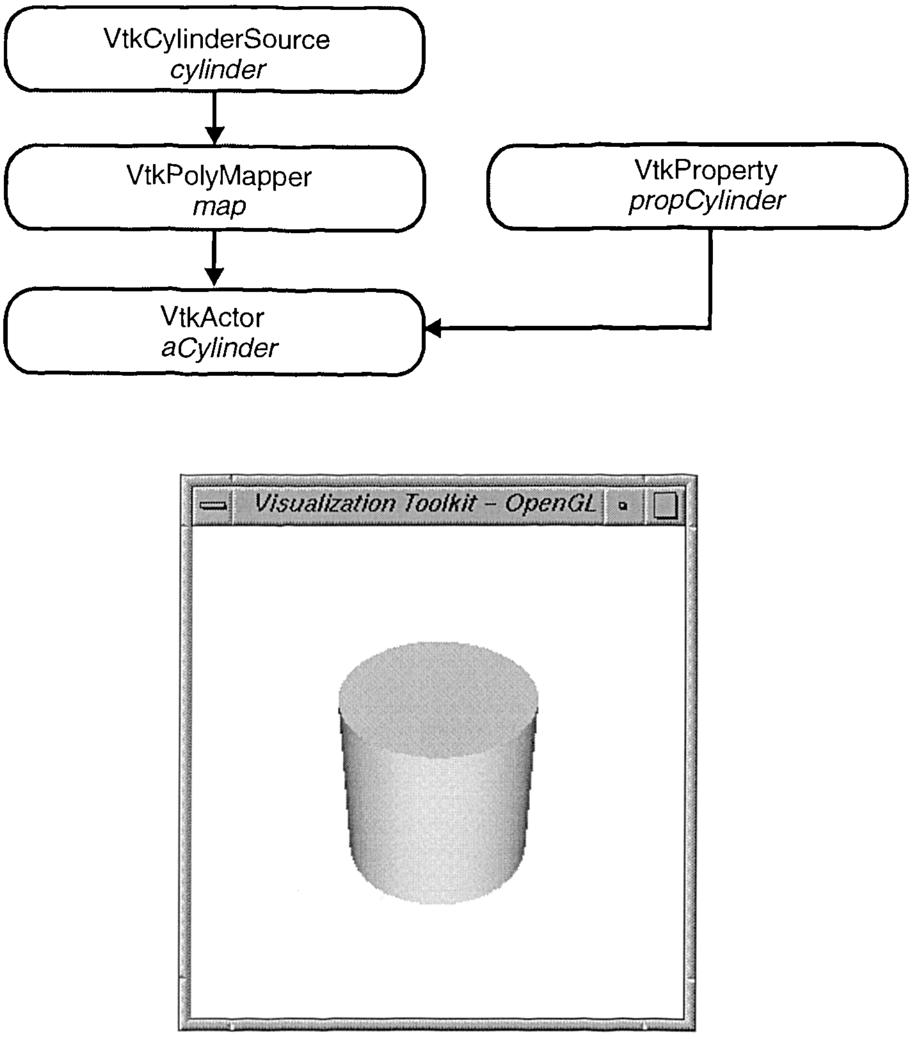


Figure 4.6 The pipeline used in the cylinder.tcl application and a screen shot

4.7.4 Integrating Tcl/ GourpKit/VTK

In order to provide all features required to develop a real-time collaborative 3D graphics application, it was necessary to integrate the Tcl language and the two main toolkits, GroupKit, and the Visualisation Toolkit, had to be integrated. The visualisation toolkit and Groupkit were inclusive of all Tcl and Tk commands, but exclusive of each other. Since these two extensions are also in the development phase, it was necessary to check conflicts between these two extensions.

A layer of wrapper code was added in the oroginal distribution of the Visualisation Toolkit source code to control the exchange of information between GroupKit and the Visualisation Toolkit. The Visualisation Toolkit instances become a command to provide the object-oriented nature. The commands create instances of their respective classes. The few methods that are not wrapped are unavailable from the Tcl interpreter.

When integrating the Visualisation Toolkit and GroupKit, some changes were made to a few foundation classes of the Visualisation Toolkit. Since information about users, such as owner and current user, is essential for objects in a multi-user environment, owner and accessibility flags were added in the vtkActor class definition. The vtkActor objects are actual graphics entities and every object in the scene was able to store information regarding user and access modes. The following shows the changes made in the vtkActor class.

```
// These functions are added for groupware features in vtkActor class defination of
// vtkActor.h.
  void SetOwnerProcess(int);
  int GetOwnerProcess();
  void SetId(int);
  int GetId();
  void SetControlStatus(int);
  int GetControlStatus();
```

A new interpreter, *vtk_gk_wish*, was created by wrapping, compiling and linking the source codes of the Tcl/Tk, GropuKit and the Visualisation Toolkit. The interpreter provided all the features of 3D graphics and visualisation, real-time groupware, basic graphical user interfaces and fundamental programmability.

4.8 Chapter summary

This chapter presents conceptual and technical backgrounds for a real-time collaborative 3D CAD system. As the starting point of a conceptual basis, the concept of shared workspace and the difference between design workspaces and general purpose shared workspaces was discussed. Typical computer based 3D design activities employed in many CAD systems were presented and issues in the development of a real-time collaborative 3D CAD system were illustrated in terms of general synchronous groupware and specific collaborative 3D features. General issues include an appropriate selection of session control, floor control, concurrency control, system architecture and awareness support. These specifically related to collaborative 3D design include:

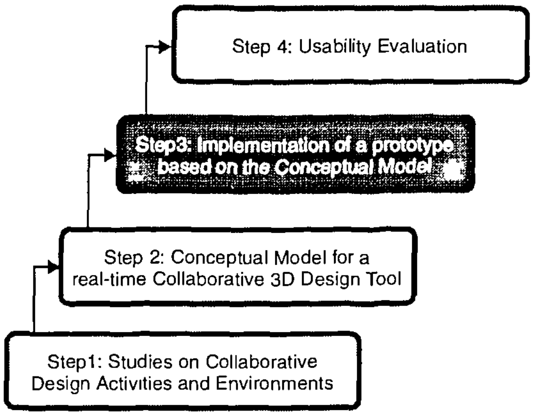
- sharing intermediate process,
- dynamic and interactive exchange of intermediate 3D data,
- support of integration activities,
- incorporation of individual and shared workspace,
- maximising acceptability of a new tool by consistent user interface and awareness support in the shared 3D workspace.

The main conceptual basis, the Shared Stage, was suggested for a real-time collaborative 3D design tool. This tool is referred to as a specific virtual 3D workspace used mainly in a collaborative situation to integrate individually created 3D components. It is both accessible and visible to all members of a team throughout a given collaborative 3D task. A conceptual software structure for the Shared Stage was also presented.

In terms of the technical background to implement the prototype real-time collaborative 3D CAD system with the Shared Stage, various software toolkits were reviewed in order to choose an appropriate prototype development environment. After a comparative review of the current collaborative toolkits, GroupKit was considered as the main development environment. Then, the Visualisation toolkit was chosen to support 3D graphics. These toolkits were customised to enable integration of the features of real-time multi-user applications and 3D graphics.

Chapter 5

A Real-time Collaborative 3D CAD System: Syco3D



5.1 Introduction

This chapter presents an evolution of a real-time collaborative 3D CAD system and details of the prototype. The implementation was primarily for investigating ways of achieving the Shared Stage framework in a software application and based on the conceptual software structure described in Chapter 4. This chapter is divided into three main sections. The first section describes the evolution of the prototype system. The details of the implementation method, results and problems in each phase are illustrated. In the second section, the user interface, collaborative features of the final prototype system are presented. In the final section, the details of the software structure are described.

5.2 Evolution of the prototype

There have been three phases of the prototype evolution. In the first phase, a real-time collaborative 3D CAD system was investigated by focusing on real-time 3D data sharing features. An existing 3D CAD application was used as an experimental system to be transformed into a real-time collaborative design tool. In the second phase, a prototype system was built using a different software structure which separates the Shared Stage module from a host 3D CAD system. In the final phase, a new 3D CAD system was explored in order to address some problems experienced during the first and second phases, and to give full control of internal data and procedures. In the final prototype system, real-time collaborative features were added to the Shared Stage Module of the software.

5.2.1 Phase 1: real-time 3D data sharing between single-user CAD systems

Before investigating data sharing features, it was necessary to determine ways of providing fundamental 3D graphics capabilities in a new prototype system. In order to reduce the amount of implementation work, it was decided to use an existing commercial 3D modelling application and to add the real-time data sharing feature. A computer aided industrial design application, Alias Studio, was selected as the experimental host system. It provides an advanced user interface for 3D design tasks and its programming tool, Alias API(Application Programmers Interface) which includes ‘openmodel’ and ‘openalias’, provides the capability of building customised functions and gaining access to the internal data structures (Alias|Wavefront, 2000).

Two Alias plug-in applications were built in order to address one of the fundamental Shared Stage features, 3D object data communication over the network between two connected Alias applications. The main function of the first plug-in application is to send local 3D object data to the other machine in real-time, while the second is to receive data from the remote machine and incorporate it into the local workspace. Figure 5.1 shows 3D object data flow during a real-time collaborative 3D design session. As shown in the figure, two plug-in applications run separately within the Alias applications. If a 3D object is created in the Alias system A, the first plug-in saves the data as a temporary file on the shared file system. The creation of a temporary file produces an event which notifies the second plug-in, in the Alias system B, which will read the data from the shared file system and automatically combine the 3D data into the local workspace data instantly. The plug-in applications were programmed in the C++ language using Alias API libraries and UNIX network socket libraries.

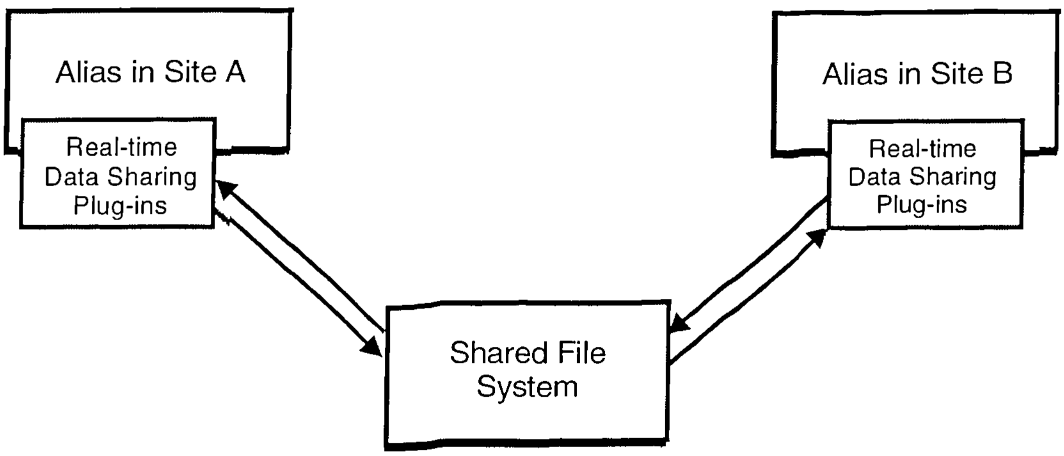
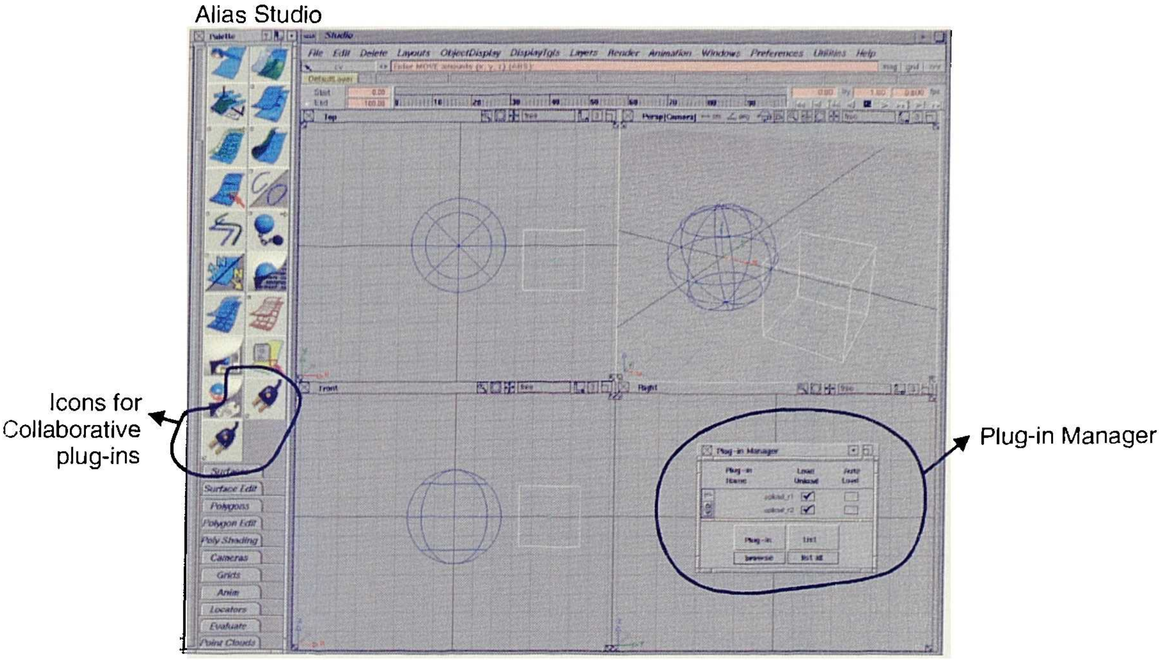


Figure 5.1 Flow of 3D object data in connected Alias systems

Figure 5.2 shows a sample screen shot whilst the plug-in applications are being launched. To use the plug-in application, the user needs to load the plug-in applications into the Alias system using a plug-in manager as shown in the figure. A successful load of the plug-in applications creates new icons in the menu palette. To activate the connection and shared status, both users need to switch on the plug-in by clicking the icons. While the plug-ins are turned on, anything the user creates is automatically duplicated in the connected Alias system. Since both users have the same 3D object data in the workspace, they can instantly observe and change objects that the other user is creating.



ıgure 5.2 A host commercial 3D design tool, Alias Studio, used in the first phase ṭ the development

he synchronous data-sharing feature allowed two users to create Alias 3D objects in p ırallel. However, a number of issues were also identified with the prototype system. Firstly, the two plug-in applications work well for construction operations, but for modification operations the replicated data could not be easily synchronised. Since information about ownership, object identification and command sequences were not replicated, the remote machine could not apply the same modification operation to the replicated object. Another issue was that once replicated objects were incorporated in a local workspace, there was no means to identify whether they were created locally or replicated because the Alias application treated the replicated objects as local objects This made it difficult to effectively organise 3D objects created by multiple people. The collaborative interaction is limited because users do not have a means to interact with each other during a parallel CAD session. Also a user's local operation can be interrupted when replicated objects are created. For example, when a user is in the middle of creating an object, the sudden addition of objects by other participants results in an interruption of the object construction operation. In order to address these issues, it is necessary to take an alternative approach of sharing real-time 3D data and processes.

5.2.3 Phase 2: a commercial 3D CAD system with an external Shared Stage module

In the first phase, the software structure of a prototype system was greatly influenced by the Alias API and plug-in environment. Although the commercial 3D CAD system reduced the concerns over fundamental 3D modelling functions and preserving the existing user interface, the functions of a prototype were considerably limited due to the data structure and software procedures of the host application. The latter is designed as a single user application.

In order to address this problem, in the second phase collaborative features were separated from the host 3D CAD system. Separating collaborative features from the host application gives several benefits. By modularising the software components according to this functionality, it is possible to reduce the complexity of the prototype development. The number of procedures and variables that need to be considered is dramatically reduced, because the collaborative parts are independent from the program routines of the host application. It also addresses several problems of reliability: such problems occur in situations where local user operations are interrupted by unexpected remote events. Specific tools for multi-user interaction can be incorporated in the collaboration module, while functions and interface of the individual workspace is preserved. For example, telepointers, workspace history information and synchronised 3D shared views are developed to support multi-uscr interaction in the external collaboration module.

This approach can be extended to a collaborative environment where heterogeneous design applications are connected via the external collaboration module, if other design applications support a mechanism to incorporate such an external module. The collaboration between users in various phases or (not only modelling but also analysis or simulation) in different disciplines may be possible using add-on standardised collaborative interaction modules that connect different kinds of applications (Nam and Wright, 1998).

The prototype at this phase was implemented as two software components: The first component, the Alias plug-in applications for interfacing 3D data between external modules and the main application, and the second component, an external collaborative interaction module. The plug-in application for interfacing 3D data between external modules and the main application was developed using the C++ language and the Alias API environment as the first phase.

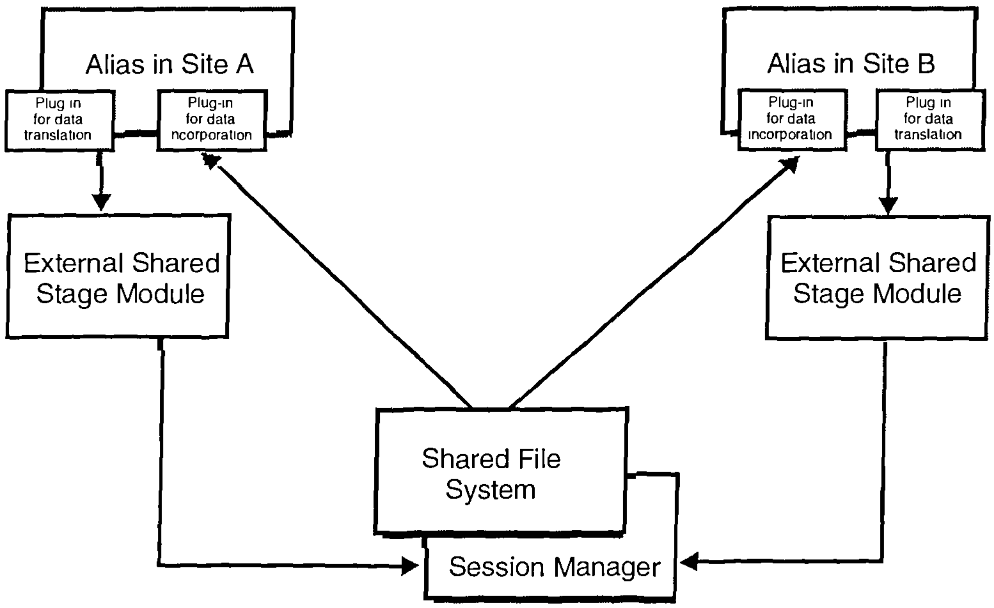


Figure 5.3 Data flow of the prototype in the second phase

Figure 5.3 shows data flow during a real-time collaborative 3D CAD session. When there is any change in a user's data within the local Alias system, it converts the Alias data to a simple geometric data format to be read by the external modules. It also saves changes of the Alias data in a shared file system. Then, the external module in the local workspace translates the data to the remote user's external module. The plug-in application in the remote Alias System automatically imports data from the shared file system. This procedure of dynamic data duplication provides an interactive and real-time exchange of 3D data between connected Alias machines.

In addition to the capability of maintaining the same status of all the data in the local and the remote workspace, the external module supports collaborative interactions. The external module facilitates integration and co-ordination tasks. For example, one of the multi-user interaction tools, tele-pointers, facilitates a co-ordination process by supporting non-verbal interaction. A synchronised camera view allows users to have a shared visual representation of the 3D workspace. Camera control is managed by a locking mechanism, so that multiple users can use the shared 3D view without any conflict. Object colours are displayed differently according to ownership. Some textual information regarding workspace events provide awareness of the activities of others. The external module is implemented using the Visualisation Toolkit, Groupkit extensions and Tcl. The group session is maintained by a GroupKit session manager called open.reg. Figure 5.4 shows an example of a real-time collaborative 3D design session using the prototype of this phase.

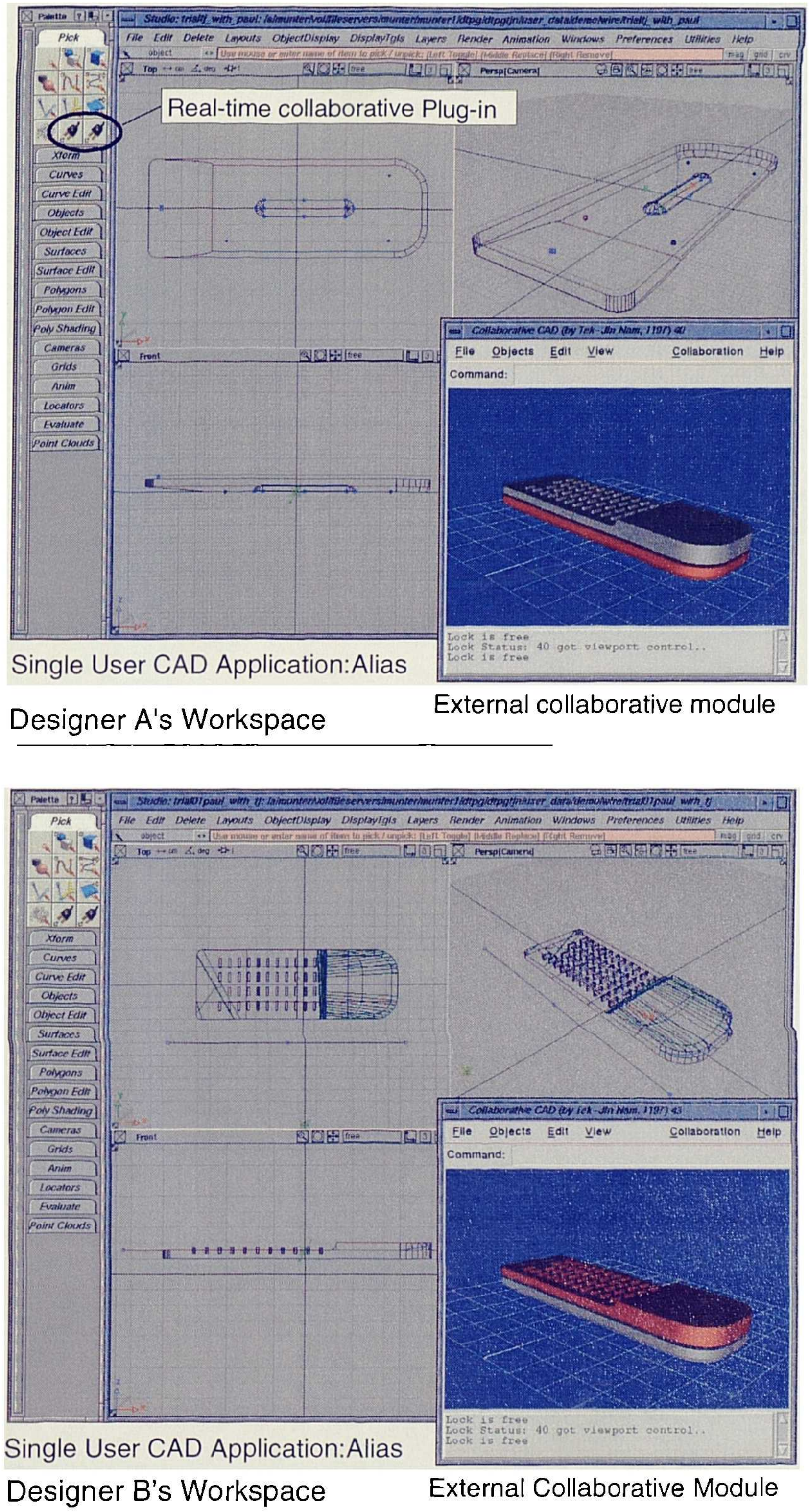


Figure 5.4 Example screenshot of a real-time collaborative 3D design session using the prototype developed in the second phase

Some technical and user interface problems were also found in this phase. One of the main problems was caused by separation of the host system and the external collaboration requiring data to be held in two separate applications. Duplicated procedures were required in both software components and with complex 3D data the performance of the system was therefore greatly reduced. In addition, there were several data format conversion processes in order to duplicate the original Alias 3D data at the same time as displaying them in the external module. Another problem was that the external module had a slightly different user interface from the host system. The inconsistency of the user interface can reduce user acceptability and make it difficult to use while the host system and the external module are being used interchangeably.

A number of Shared Stage features were achieved in this phase. However, because the implementation was still dependent upon the programming tools of the host system which was a single-user oriented application, the overall result was a limited extension of single user application. By controlling and monitoring all command sequences, more sophisticated collaborative features, such as dynamic data communication, various multi-user interface tools and collaborative modification of 3D objects in a multi-user environment could be achieved. In order to investigate the full potential of a Shared Stage system, it became necessary to build a new CAD system allowing full control of the software structure.

5.2.3 Phase 3: a new 3D CAD system with the Shared Stage module

The use of a single user oriented commercial 3D application as a host application reduces the effort of implementing a number of 3D graphics functions, but limits ways of supporting collaborative features in the first and the second phase. In order to address issues raised in the previous chapter regarding the real-time collaborative 3D design tool (Chapter 4.4), it is necessary that a collaborative tool has data structures and procedures that can represent multi-user aspects. Therefore, in the final phase, a new 3D design tool was built to consider these aspects integrating a collaborative module as a standard component in the application.

Before focusing on the real-time collaborative features of the Shared Stage module, a set of fundamental functions for 3D CAD tasks were identified through the review of common features of current CAD systems. Basic functions were selected, taking into account users abilities, in order to complete a usability evaluation which will be described in Chapter 6. The fundamental functions were grouped into menu and tool palette functions. A standard menu-driven user interface and 3D interaction technique using multiple projection views and mouse was adopted, to provide a familiar environment.

The Shared Stage module was then implemented as a light synchronous groupware application that allows users to share views and structural representations of the workspace. Various collaborative features were investigated and incorporated in the Shared Stage module. Once the single user 3D CAD system and the stand-alone Shared Stage module were built, a connection between the two was made.

Objects created in a local workspace are processed in several steps, in order to facilitate sharing with other users. Objects are built and replicated in the Shared Stage module, then all users in a group session can access the objects through the shared synchronised camera view of the Shared Stage module. When objects are created or modified, events are created to notify the local Shared Stage module. A copy of these objects is then sent to the remote systems from the local Shared Stage module. The replicated 3D data is automatically shared by all connected Shared Stage modules in a collaborative session. During the replication procedure, the Shared Stage module only updates the modified part of the model in order to make the replication process more efficient. On the other hand, any meaningful changes in the Shared Stage module are reflected in the local workspace. For example, when the ownership of an object changes, an event is generated to notify it to the local host application. The local host application updates the private workspace according to the instruction transferred from the local Shared Stage module. Changes in all workspaces and creation of objects are reflected automatically in this manner.

Details of the user interface, collaborative features and software structure of the final prototype system of a real-time collaborative 3D design tool, Syco3D, are presented in the following sections (section 5.3, section 5.4).

5.2.4 Consideration of real-time collaborative 3D design issues in each phase

During implementation, the issues identified in the previous chapter (Chapter 4.4) were considered to a different level. In the early phase, the essential features of the collaborative application are the focus, while in the later phase most of the issues are addressed. Table 5.1 shows major issues considered in each phase of the prototype evolution.

Table 5.1 Consideration of issues of real-time collaborative 3D design tools in each phase

	Phase 1	Phase 2	Phase 3
General requirement			
Session control	X	O	O
Floor control	X	O	O
Access control	X	O	O
Concurrency control	X	O	O
Awareness support	X	<>	O
System architecture	Replicated	Replicated	Hybrid
Requirements specific to 3D design			
Sharing intermediate processes	<>	<>	O
Dynamic and interactive exchange of intermediate 3D data	O	<>	O
Support of collaboration for integration activities	X	<>	O
Incorporation of individual and shared workspaces	X	<>	O
Maximising acceptability of a new tool by consistent user interface	O	O	O
Awareness support in the shared 3D workspace	X	<>	<>

X : not considered, <> : Partially considered, O : fully considered.

5.3 A real-time collaborative 3D CAD system: Syco3D

5.3.1 Overview

Syco3D represents Synchronous Collaborative 3D CAD. It is a prototype of a 3D CAD system allowing users to create and modify 3D models collaboratively in real-time. Syco3D is not a fully functional 3D CAD tool, but has sufficient functionality to build and modify 3D models using primitives. As a multi-user system, Syco3D provides a shared 3D workspace called the Shared Stage Window in which users are able to accomplish joint 3D CAD activities having various multi-user interaction support.

5.3.2 Basic 3D CAD features of the system

Basic 3D modelling functions are grouped into the menu and tool palette functions. The menu consists of the following function groups.

- File – functions for saving and retrieving files and exiting the system.
- Edit – functions for editing: copying, pasting, grouping and ungrouping.
- Delete – functions for deleting.
- ObjectDisplay – functions that allow object display types to be selected.
- Views – functions that create new 3D views.
- Multiuser – functions for multi-user interaction, including opening the Shared Stage Window and synchronising the status of the shared workspace.

The tool palette includes commands that have direct interaction with the views. The functional groups in the tool palette are as follows.

- Pick – functions for picking objects in the modelling workspace
- Xform – functions for transforming objects: moving, scaling and rotating
- Objects – functions for creating and locating primitives

Figure 5.5 shows an example Syco3D layout which shows the arrangement of the menu and tool palette.

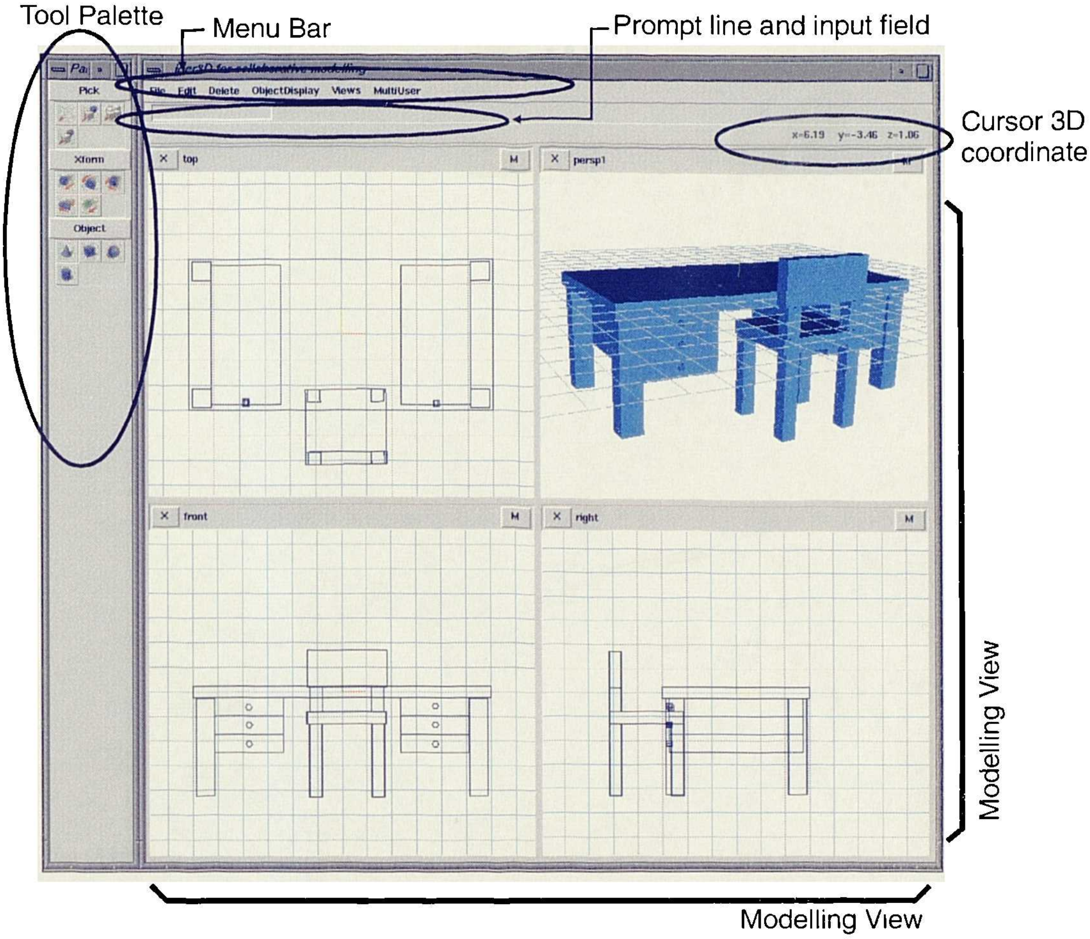


Figure 5.5 Screen layout showing the functional groups of the menu and the tool palette of the Syco3D system

5.3.3 Collaborative features of the Shared Stage Window

In addition to basic functions of 3D CAD, the Syco3D system provides a shared 3D design workspace, Shared Stage Window, where users in a real-time collaborative session can collaborate on 3D design tasks. The Shared Stage Window can be turned on or off by choosing one of the menus in the MultiUser menu item. Figure 5 6 shows the layout of the Shared Stage Window which includes the Synchronised Stage View (SSV) and Data Structure Diagram (DSD).

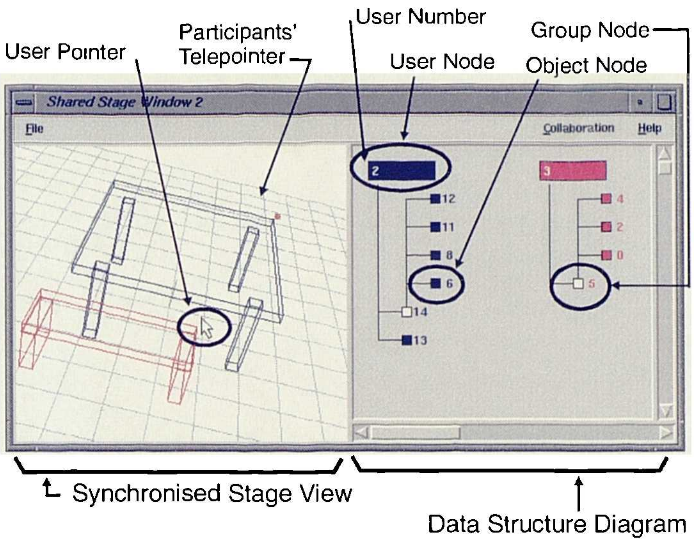


Figure 5.6 Layout of the Shared Stage Window

.3.3.1 Synchronised Stage View (SSV)

SV provides information about how a user's 3D work is related to others through a ynchronised shared 3D representation. The main collaborative features of the SSV are s follows:

- *Synchronised shared 3D view*: All users in a session see the same 3D view of a 3D virtual workspace and any change in the view is updated instantly When the independent review of the shared workspace is preferred by users, the perspective camera view in the private Syco3D window and the dynamic data referencing feature of the DSD is used.
- *View control co-ordination*: Multiple users can control the 3D view with minimum conflict. One person at a time controls the 3D view with a locking mechanism. When other people wish to control the view, visual cues are generated to notify the view control request.
- *Multi-user cursor and representation of ownership*: A user's individual pointer is displayed in the owner's colour, which is assigned when the Syco3D system starts.

This allows each user to point to location in the Shared Stage module, supporting gestural communication. All objects in the Shared Stage module are also displayed in the owner's colour, allowing users to recognise ownership information of the objects.

- *Real-time update of workspace changes*: Any change in the individual workspace is updated instantly during a collaborative session. By observing the SSV, one can be aware of the progress of others' work.

5.3.3.2 Data Structure Diagram (DSD)

DSD provides structural information about the contents that change during a collaborative 3D CAD session. The information is represented in a tree-like diagram, which consists of different types of nodes. Figure 5.7 also shows a sample tree with three basic components: the user node, the group node and the object node. A user node represents a user through user id and colour. Object nodes correspond to 3D models in the workspace. Group nodes represent grouped objects, such as an assembly of parts. The first tree on the left shows one's own model structure. The subsequent trees placed to the right represent model structures of other users in a collaborative session. The following features are available in the DSD.

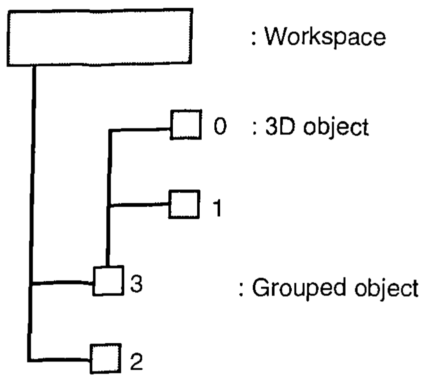


Figure 5.7 Components of the Data Structure Diagram

- *Representation of workspace past changes*: Recent changes of data are represented at the top of the tree diagram. This indirectly shows workspace history information, such as recent creations or editions.
- *Presenting users' status information*: A user node is automatically created when a new user joins a session. The representation of a user node is changed to provide status information of the user. For example, when a user controls the SSV or is waiting for a view control, the corresponding user node is highlighted or blinked.
- *Real-time data transfer and referencing*: Each user can transfer models across to others' workspaces in real-time by dragging an object or user node onto their user node. Transferred models can be duplicated or modified directly. Users can also reference the models created by other people in order to use them as a template without making any change on them.
- *Resolving problems caused by simultaneous access to the objects by multiple users*: Since users do not make changes directly in the Shared Stage but only in their private Syco3D windows, the potential problems caused by simultaneous access to objects by multiple users are avoided.

5.3.4 User scenario

The following scenario illustrates the user interface of the real-time collaborative features explained above. This scenario may take place either in a co-located or in a distributed situation. A group of three designers are working together in real-time to construct a virtual 3D model of a vehicle. After a short discussion, they decide that the body frames for the vehicle should be modelled by one designer, exterior parts including bumpers and tyres by another designer, and interior parts including seats and a steering wheel by the other designer. At the start of the collaborative session, designers choose their user colours exclusively. One designer chooses green, another chooses blue, and the other chooses red. User identification numbers, 2, 3, and 4, are assigned by the session manager.

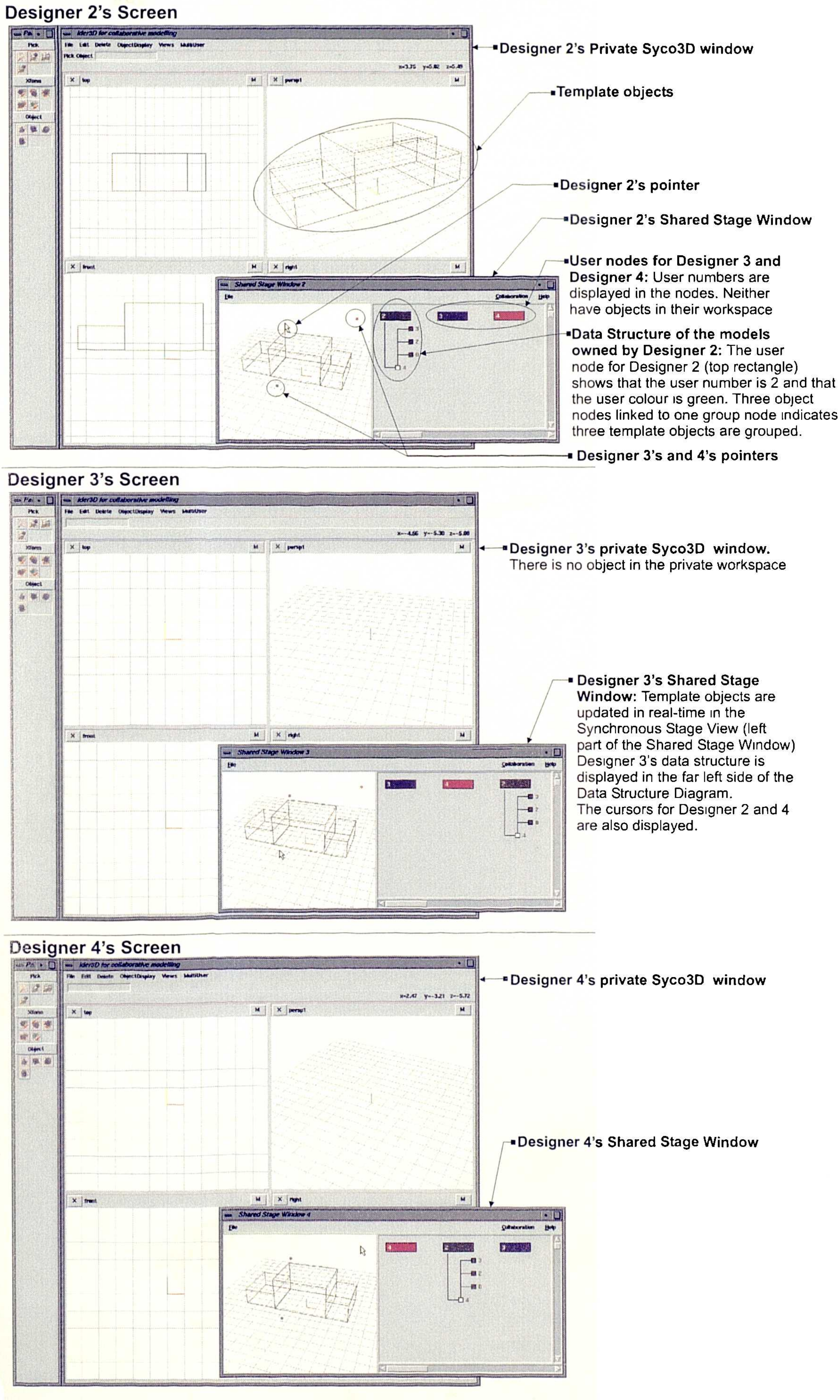


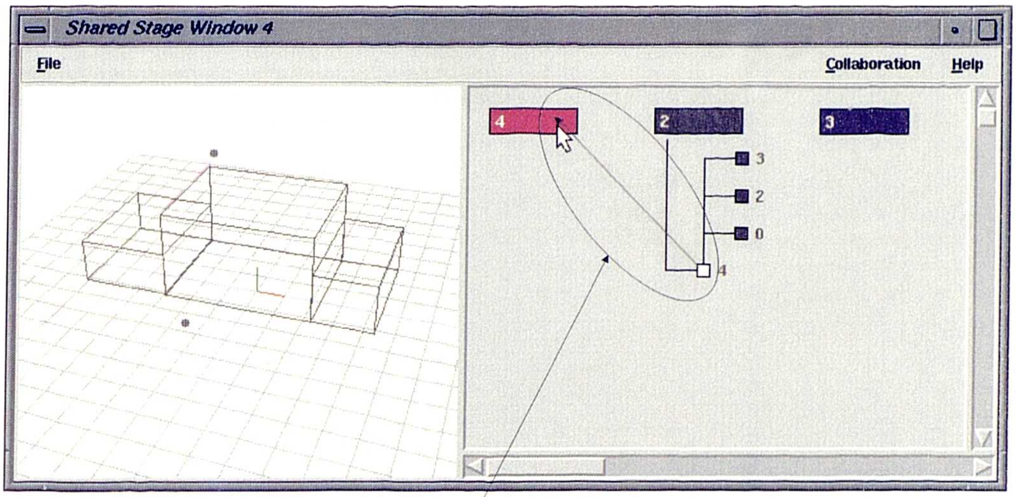
Figure 5.8 Screen shots in three different sites in the early phase of a real-time collaborative CAD session for modelling a vehicle

Figure 5.8 shows screen shots from each designer's computer in the early phase of the collaborative session. Each designer has a private Syco3D window and the Shared Stage Window. Template objects are created by Designer 2 to share modelling parameters, such as scale and location of parts in the workspace. Other designers observe the creation of the template objects instantly through the Synchronised Stage View.

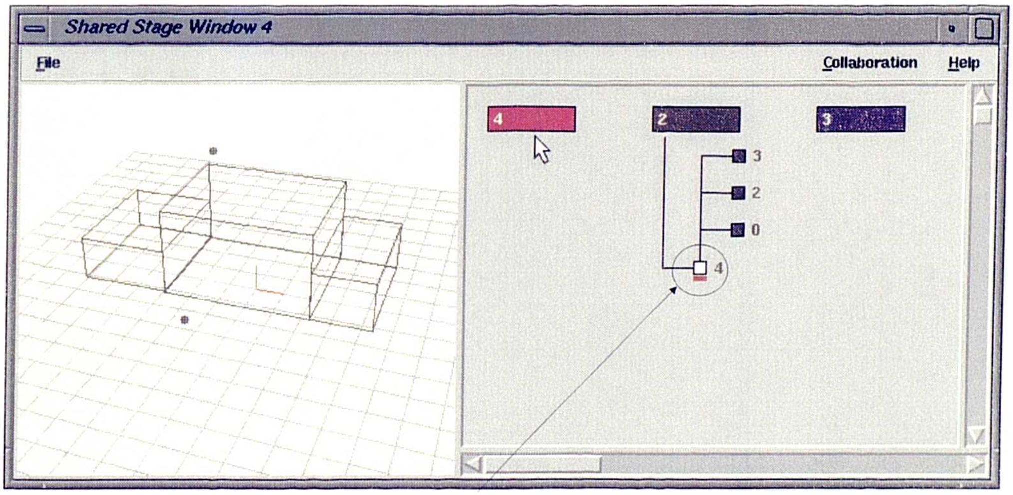
The template objects can be displayed as a reference in Designer 3 and 4's private Syco3D windows by dragging the nodes of corresponding objects onto their own user nodes. While dragging, an arrow is displayed to visualise the operation of referencing (Figure 5.9). Designer 3 and 4 can see the template objects in their private windows but cannot modify them. The referencing status of the object is represented underneath the corresponding nodes by a small rectangle in the colour of the user who is referencing the object. When more than one designer uses the object as a reference, multiple rectangles are displayed.

The three designers construct 3D models of allocated parts concurrently. They often shift their focus between their private Syco3D window and the Shared Stage Window. Designers exchange ideas about modelling methods and the details of parts interacting in the Shared Stage Window.

View control is co-ordinated by allowing one designer control the view angle of the Synchronised Stage View at a time. The control status is displayed as an outline of the user node. When other designers try to manipulate the view with the mouse in the Synchronous Stage View, the outline of the corresponding user nodes is blinked as a visual cue to notify the request (Figure 5.10). Verbal interaction and an audio feedback from the system are also used to co-ordinate this view control of the Synchronised Stage View.

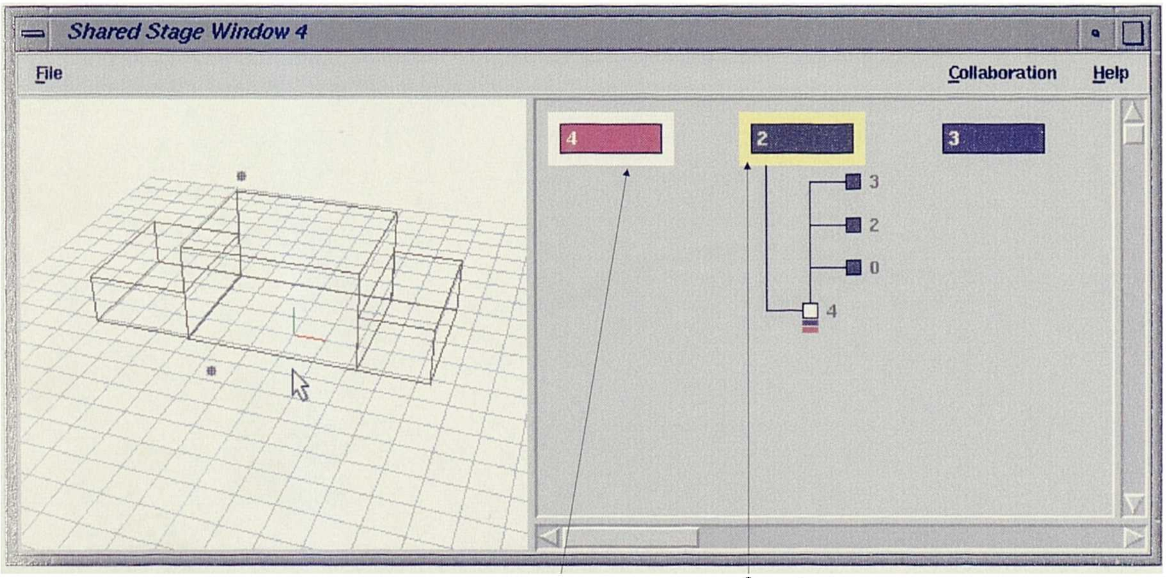


▪ Referencing operation is carried out by dragging the node onto the user's own user node with the middle mouse button. An arrow is displayed while dragging.



• **Reference status of the template objects**
Rectangles are displayed under the nodes in user colours to display the referencing status of the object. Multiple rectangles may be displayed when the object is referenced by more than one designer.

Figure 5.9 User interface of referencing operation



Outline showing current user controlling the view of the Synchronised Stage View

Blinking outline showing this user requests the view control

Figure 5.10 Co-ordination of view control among multiple designers

Figure 5.11 shows screen shots from each designer at an advanced phase of the vehicle modelling project. The scene becomes more complicated as more parts are added in each workspace. Designers may employ two methods to reduce the visual complexity in the Shared Stage window: representation by user colour and simplification of object representation. In the Shared Stage Window, models and the corresponding objects are displayed in the same owner's colour that is assigned at the start of the session. This helps designers understand the scene by organising the parts by their ownership. The Shared Stage Window can be simplified to reduce the complexity of the scene. For example, the branches of the model data tree are hidden by double clicking the user node. This makes the representation of the corresponding objects simpler by representing it as a bounding box. (Figure 5.12).

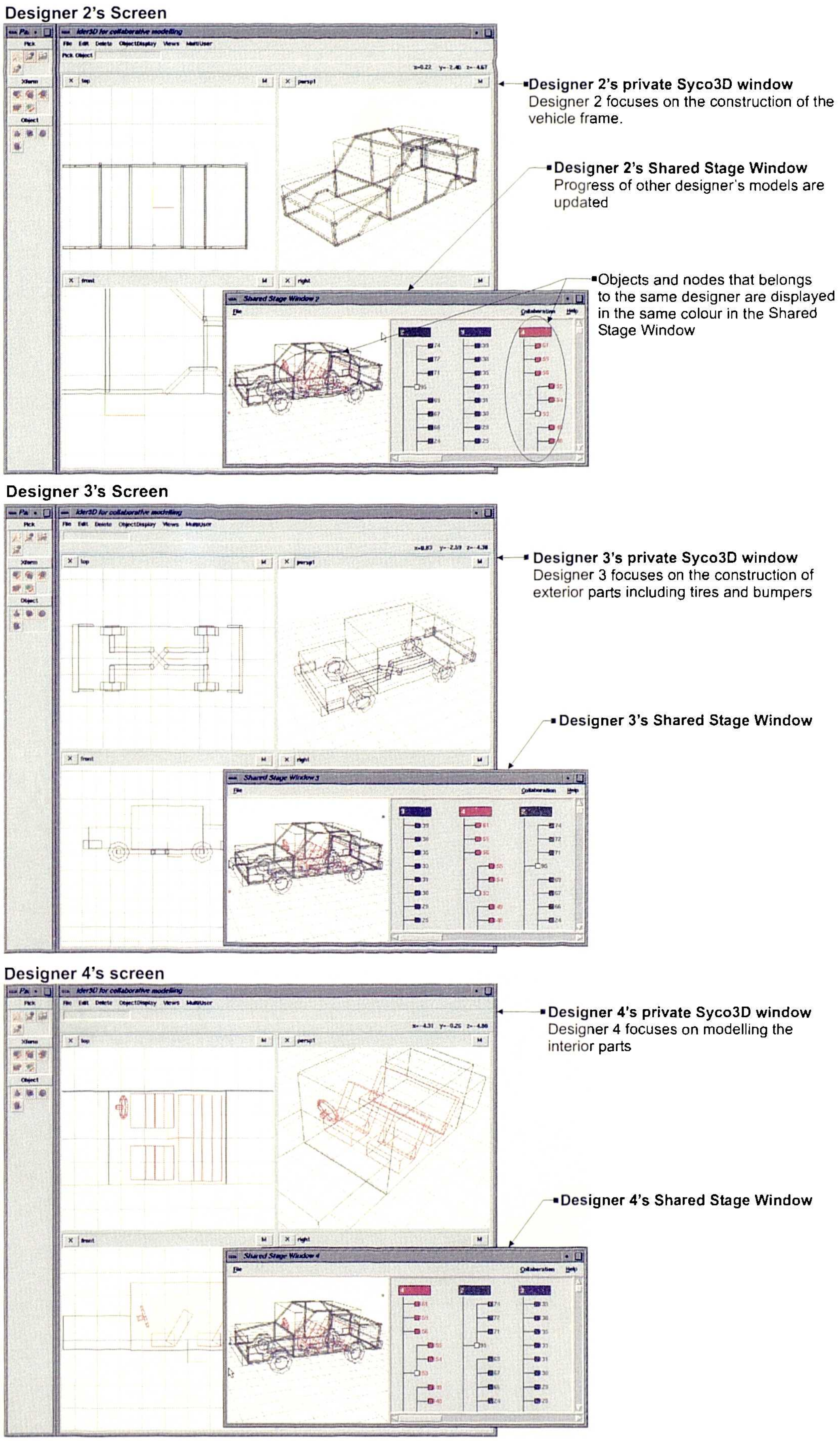


Figure 5.11 Screen shots in three different sites in an advanced phase of a real-time collaborative CAD session for modelling a vehicle

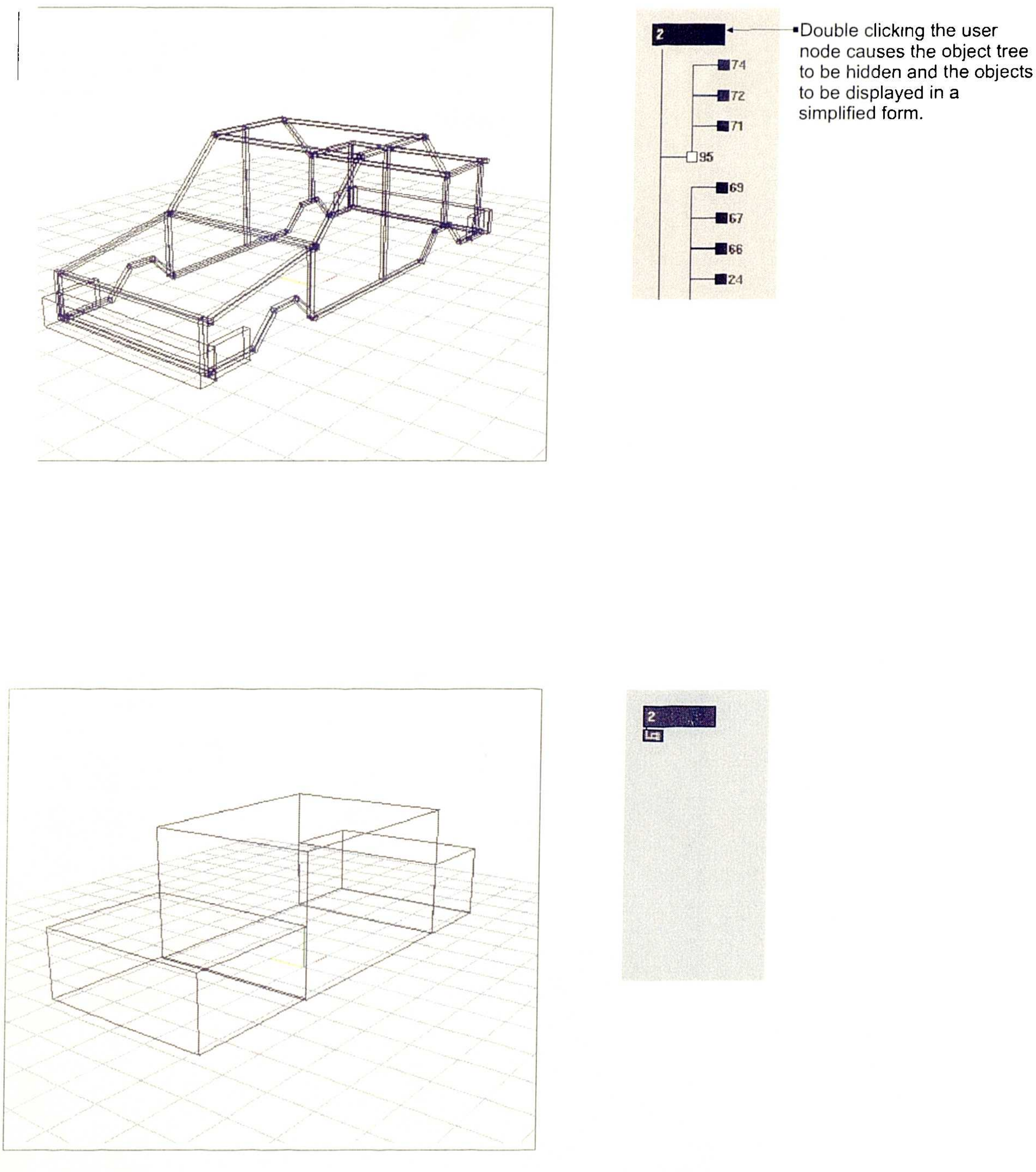
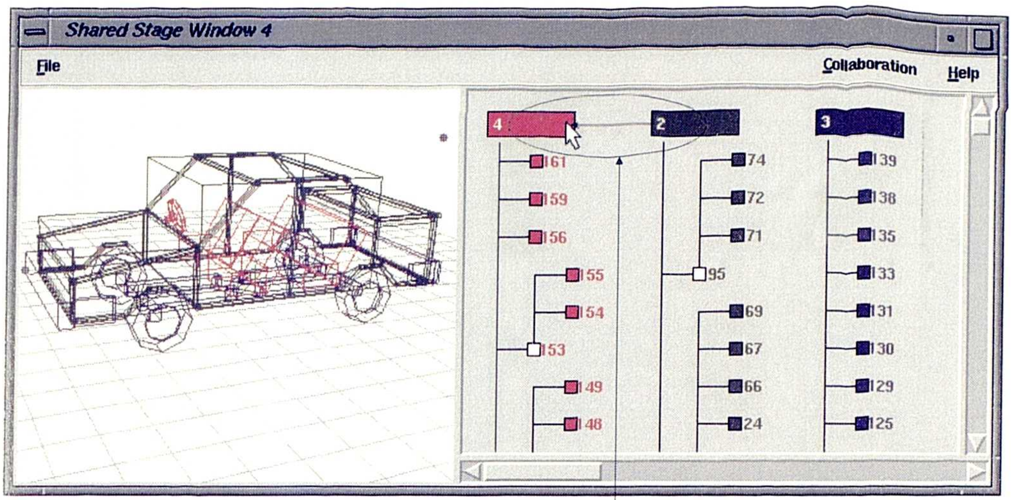


Figure 5.12 Reducing visual complexity using simplified representation

One can check entire models created by other designers independently in their private Syco3D window. Referencing entire models is carried out by dragging the corresponding user node onto his/her user node (Figure 5.13). Although the designer sees the referenced models in his/her private Syco3D window, modification is not allowed. Objects are searched and checked in the Shared Stage Window by selecting the nodes in the Data Structure Diagram or objects in the Synchronised Stage View. When nodes or objects are pressed, the corresponding objects or nodes are highlighted (Figure 5.14).

Designers may transfer models in real-time across different users' private Syco3D windows. Transferring is carried out in a similar method to the referencing operation but with the left mouse button. For example, if designer 2 completes the frame modelling and wishes to help designer B build bumper objects, Designers 2 brings the front bumper object from the designer C's data tree. Then Designer 2 copies and modifies the object to create a rear bumper. When objects are transferred across different workspaces, the confirmation window is displayed during the operation (Figure 5.15).



Designer 4's screen after referencing entire models freated by Designer 2

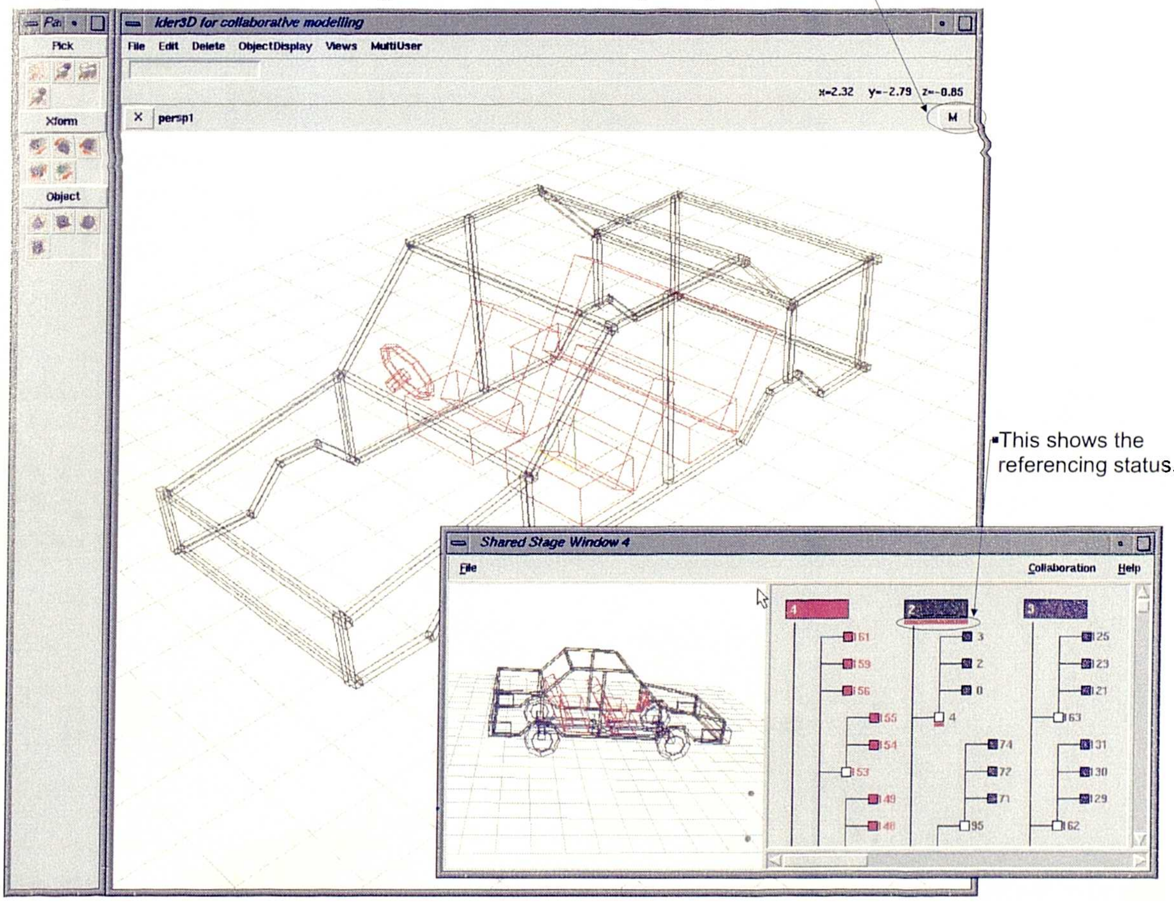
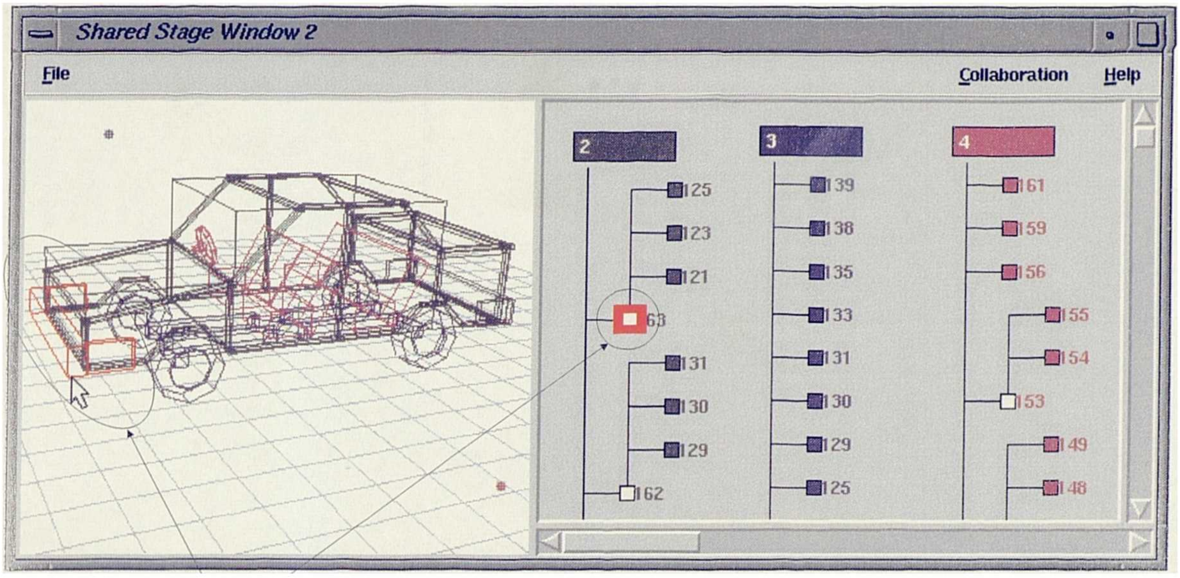


Figure 5.13 User interface for referencing entire models created by a user



If a model or a corresponding node is clicked, both are highlighted for ease of search

Figure 5.14 User interface for reviewing objects

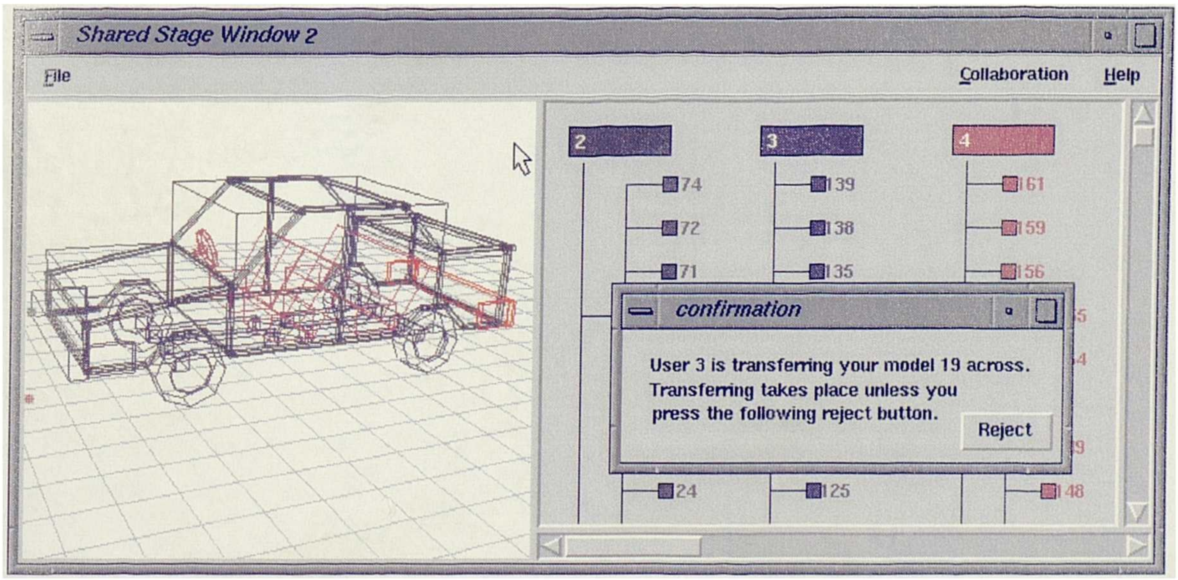


Figure 5.15 Confirmation message while transferring a model

Sometimes designers help each other to solve technical problems relating to the creation of virtual 3D models. This also supports user awareness of progress in the modelling process. When one designer makes a significant change influencing the modelling parameters of others' parts, the other designers are instantly notified of the change by visual feedback in the Shared Stage Window.

When all parts have been modelled, designers can integrate completed parts collaboratively. One designer brings all the parts into his/her private Syco3D window. Incorporation into one workspace can then be achieved by dragging user nodes with the left mouse button. While one designer integrates and finalises the entire model, other designers provide feedback about the completed model through the Shared Stage Window. Figure 5.16 shows the completed vehicle model constructed in the Syco3D system collaboratively by three designers and integrated into Designer 2's workspace. As all designers share the modelling process concurrently and know how the completed model has evolved, few modifications are necessary in the final stages of the modelling process and a higher level of user satisfaction can be achieved.

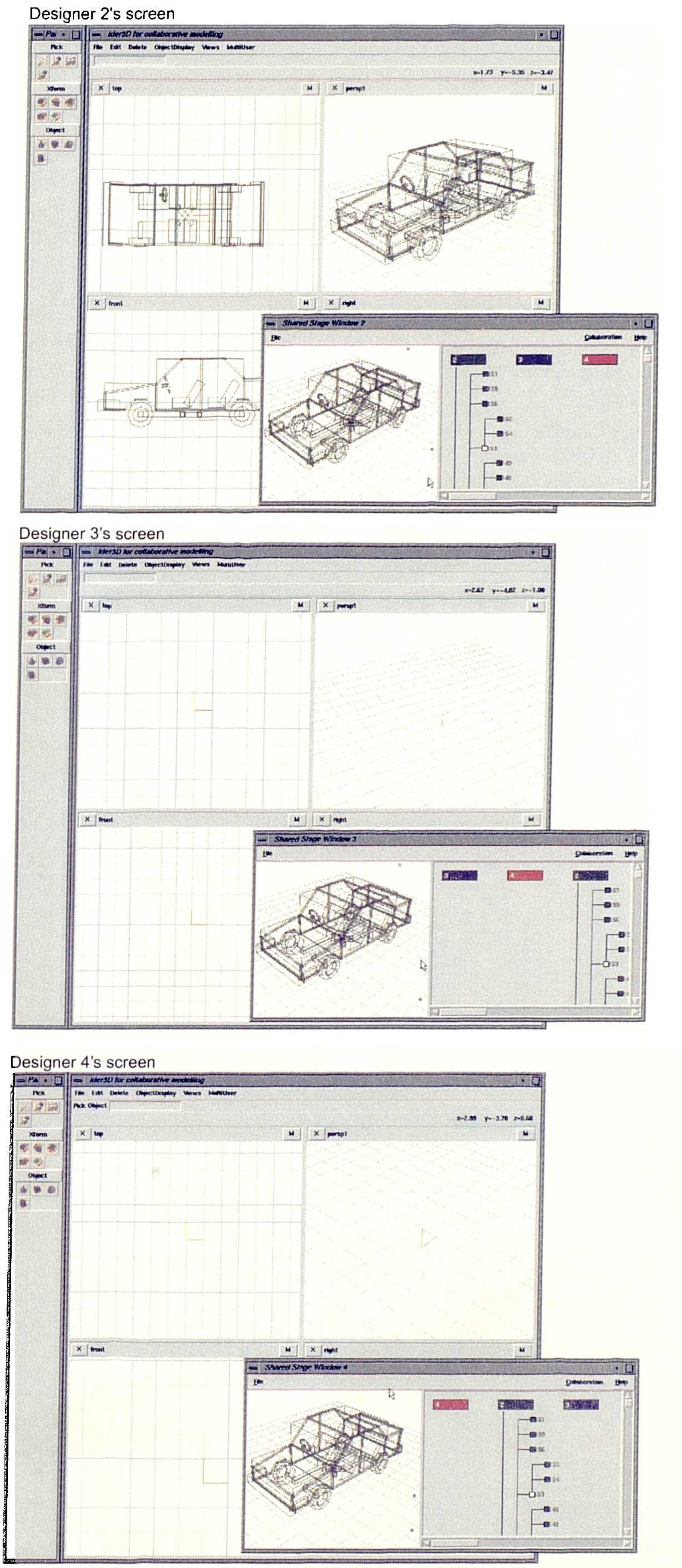


Figure 5.16 Screen shots in the final phase

5.4 Software structure

As shown in Figure 5.17, Syco3D was coded using the Tcl/Tk script language (Ousterhout, 1994) under a UNIX operating system, IRIX6.5. Tk extension was used to construct basic interface components, such as menu, prompt, icons and so on, while the Visualisation Toolkit (Schroeder et al. 1996) to implement 3D graphics features. An introduction to these tools and the way of integrating them are given in the previous chapter (Chapter 4.6).

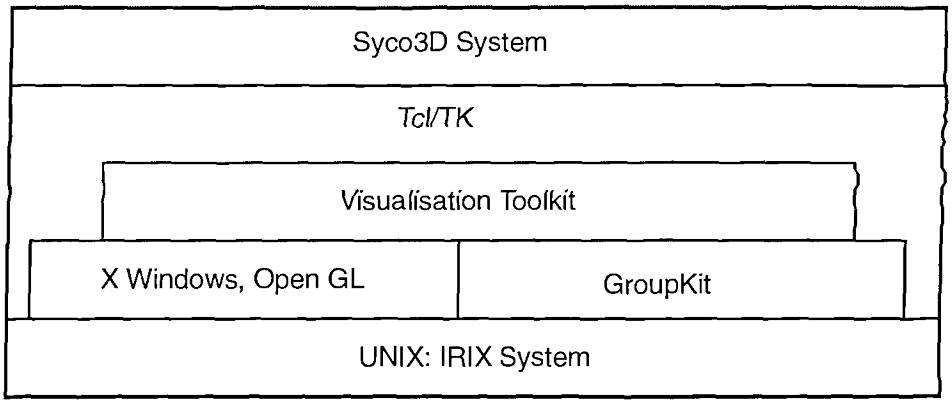


Figure 5.17 Software Development environment for the Syco3D system

Figure 5.18 shows command message flow between the Syco3D systems and GroupKit's communication infrastructure to support conference management. The communication infrastructure has a hybrid structure of server-client and peer-to-peer. The registration application of the GroupKit, *registrar*, runs as a server, and the *open.reg* session manager application acts as a client. The registration application controls the communication between the *open.reg* session managers. The Shared Stage module of the Syco3D system is launched via the *open.reg*. When launched, a connection is made to the host Syco3D system. The Shared Stage module co-ordinates the duplication of commands and data from the host Syco3D system. It communicates with the parent session manager being a part of the client system. The Syco3D processes also communicate with each other through TCP/IP sockets that connect every pair of machines. The exchange of messages are accomplished by the features of Remote Procedure Call and the shared environment features.

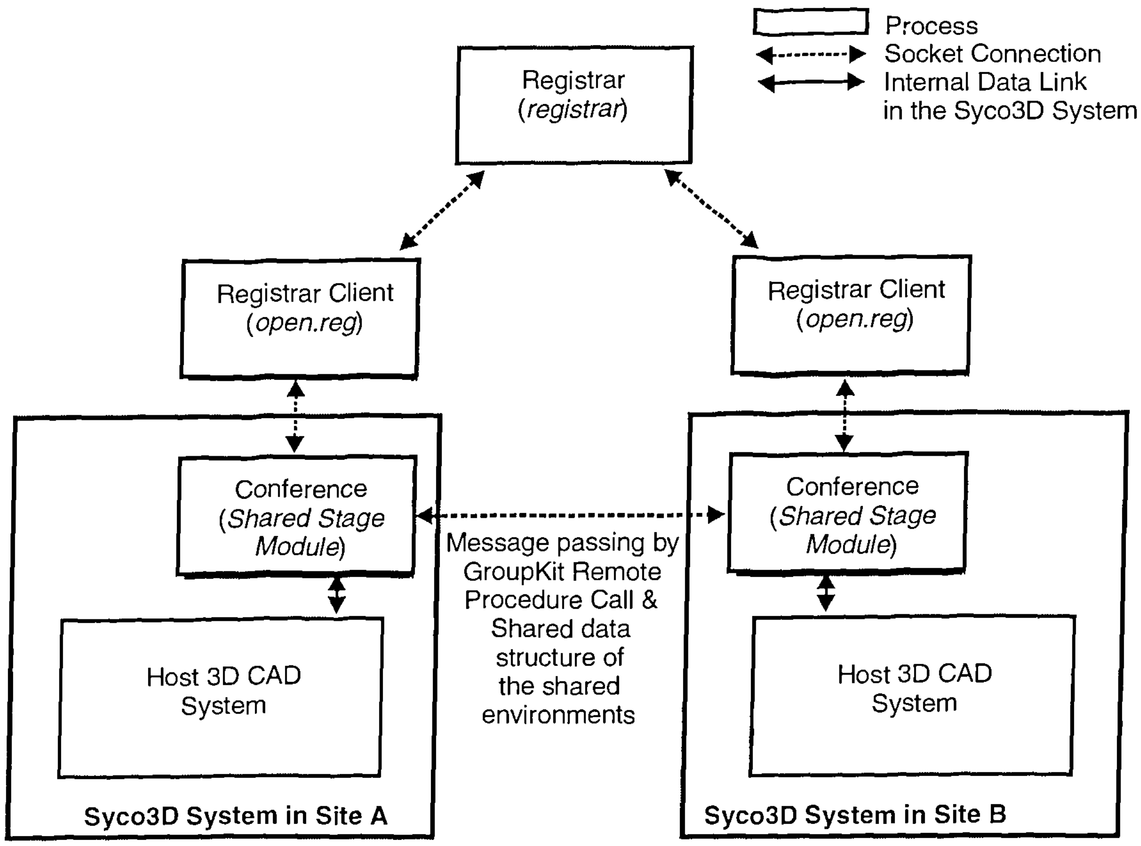


Figure 5.18 Message flow between Syco3D systems and GroupKit's communication infrastructure

The object data within the Syco3D system is structured using various classes of the Visualisation Toolkit. The *vtkActor* class of the Visualisation Toolkit is used for individual 3D objects and the *vtkActorCollection* class for a group of 3D objects. A member function of the *vtkActorCollection* instance provides a method of searching and accessing to all the objects in the group list. The list of selected objects are maintained as a *vtkActorCollection* instance. This list of current objects is closely linked for transformation functions. Other function groups, such as group, copy and paste, also use a selection list of objects. The database within the Syco3D system can be represented as in Figure 5.19. When users move data across workspaces, the objects or groups are removed from the database list of the source workspace and created in the list of destination workspace.

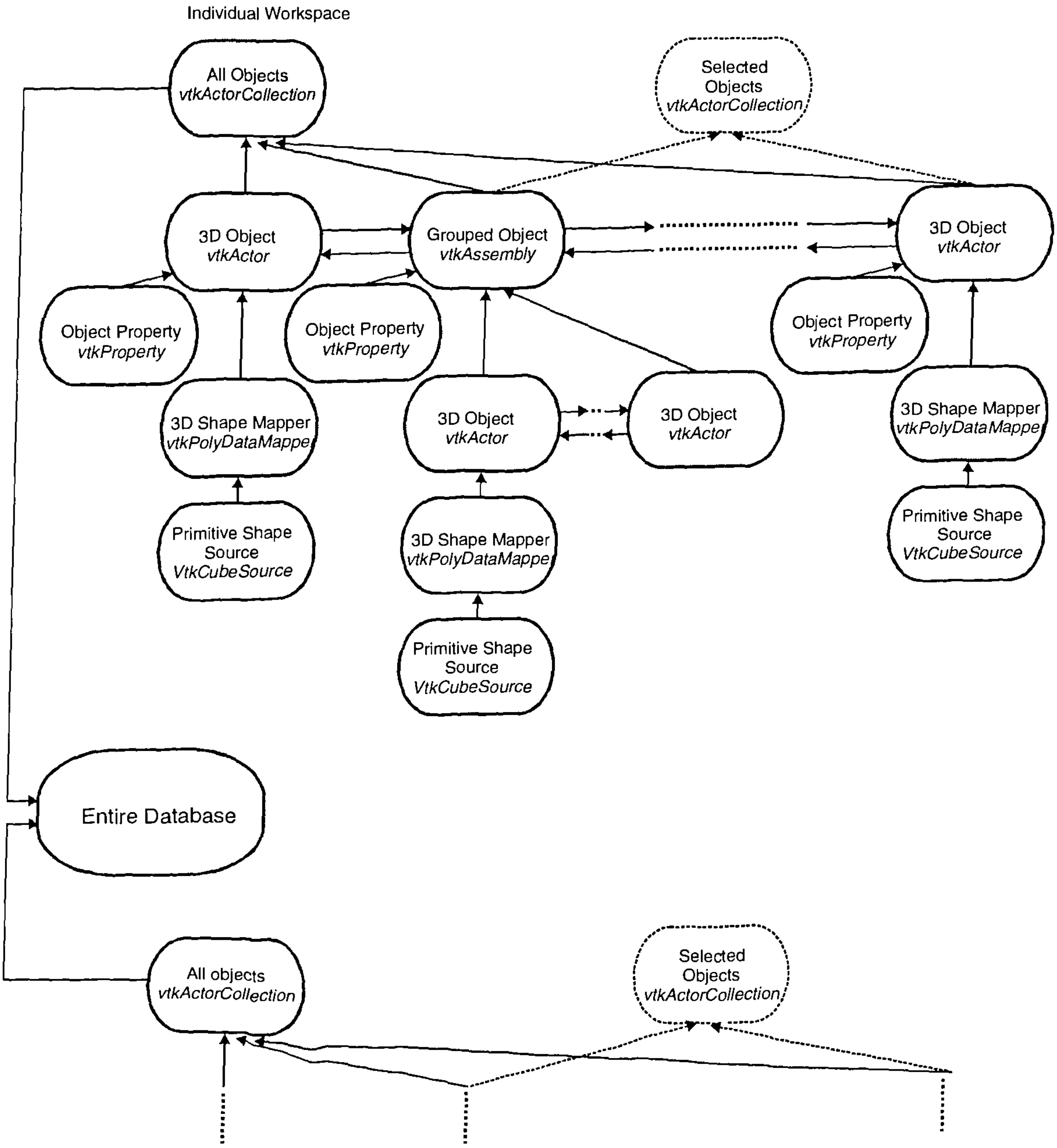


Figure 5.19 Data structure in the Syco3D system

Basic procedures required to provide the essential function groups are built in two levels to support the Shared Stage features. The procedures of the first level are executed locally, in a host component of the system, without affecting other remote systems. The procedures of the second level are replicated via GroupKit's RPC(Remote Procedure Call) commands so that remote systems execute the same procedures remotely. The procedures in the second level are those that directly relate to the change of CAD database, such as creation and modification of 3D objects. The GroupKit shared data structure, *shared environment*, was also used to synchronise the status of the Shared Stage Window. For example, the parameters of the synchronised shared view are kept as a parameter of the shared environment. They are replicated and synchronised automatically when one changes those parameters by changing the views.

Figure 5.20 shows a state diagram of the Syco3D system, illustrating the user interface of the Syco3D system. The ovals in the diagram show a state, the arrows show a transition from one state to another and the labels on the arrows show an event that causes the state transition. Each state involves several Tcl procedures displayed in the oval shapes. When the Syco3D system is launched through the session manager, it initialises local and remote databases and variables essential to multi-user environments, such as user identification and associated interface instances. Then the system waits for command input from the user. Syco3D has several function groups. Each function group is activated by selecting menu items or icons. Among these procedures, second level procedures directly modify the database are broadcasted to other Syco3D systems that are connected. As shown in the figure, grey ovals indicate a Tcl procedure that is invoked in other machines. These Tcl procedures send user information as an argument to allow other systems to accurately update the remote database. In completion of each function group, the system updates the render windows in both the base program and the Shared Stage Window.

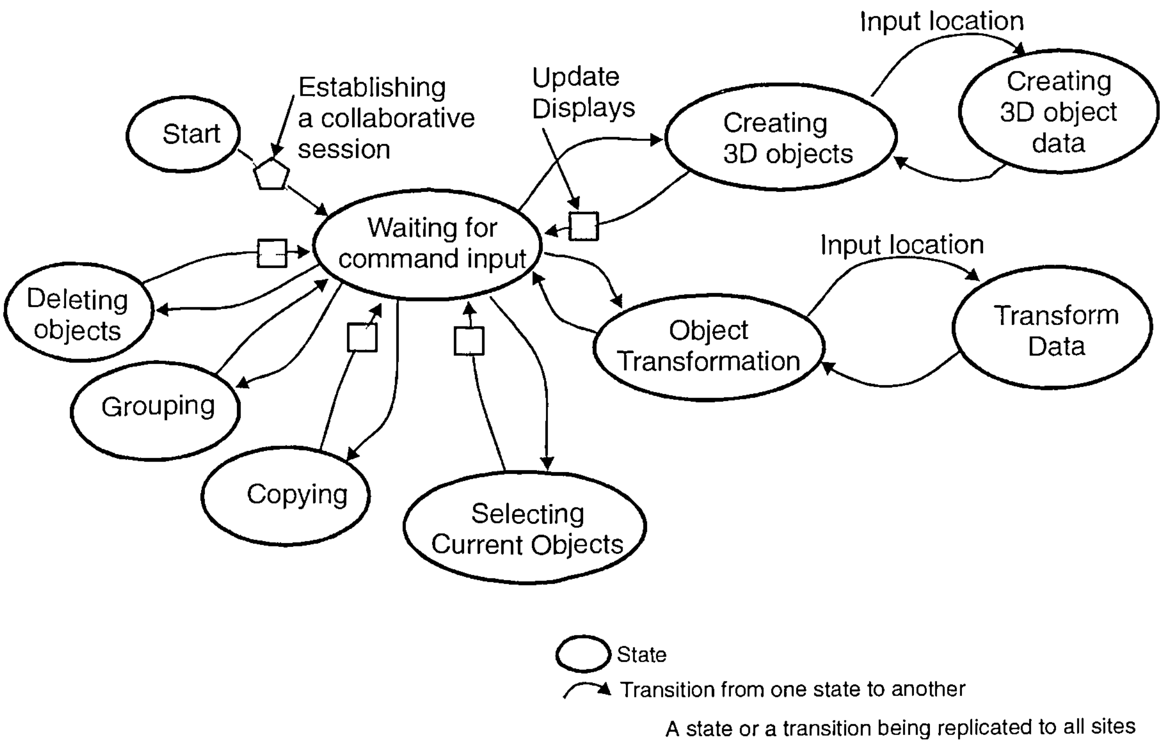


Figure 5.20 State Diagram for the user interface of the Syco3D system

Tcl procedures executed in remote systems are also invoked in the local system so as to update the remote database. When changes are made in the database, related events are also generated to update and synchronise the views of the Shared Stage Window in both local and remote sites. The source code for the Syco3D system is presented in Appendix C.

5.5 The result of the prototype investigation

Through the evolution of the prototype, Syco3D, which incorporates the Shared Stage, is built as an example software system for real-time collaborative 3D CAD. A number of collaborative features which are customised for 3D CAD are incorporated within the collaboration module of the system.

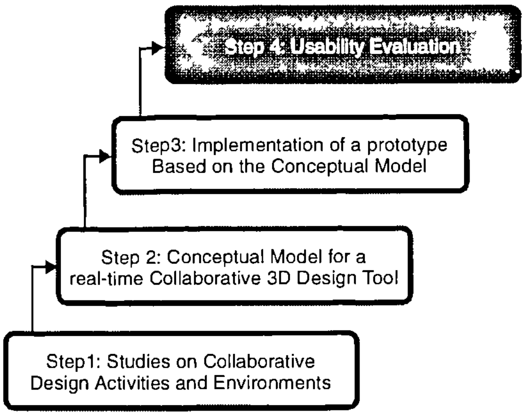
Although the Syco3D system is a simple test-bed 3D CAD application, it is possible to create adequate 3D models. The Remote Procedure Call and the shared environment of GroupKit are effectively used to address many problems, which are experienced when the single-user oriented commercial 3D application is used. It provides higher level programming abstractions for collaborative features. The approach used during the software prototype evolution showed a new strategy for developing real-time collaboration support to sophisticated computer based 3D design tasks.

5.6 Chapter summary

This chapter described the evolution of the prototype real-time collaborative 3D CAD system and the resulting prototype software. The evolution involved three main phases. In the first phase, in order to provide real-time data sharing, an existing 3D CAD system, Alias Studio, was modified. Although two separate systems could exchange 3D data via a direct local network connection, problems were experienced in reliability and in the alteration of remote objects. An alternative approach was taken in the second phase, in which the module of the Shared Stage is separated from the host system. In this modularised situation, dynamic integration of both individual and collaborative features was achieved. However the implementation was still limited to the host commercial system. A new 3D CAD system, Syco3D, was therefore implemented. Various collaborative features, such as synchronised workspace view, muilti-user telepointer and dynamic data exchange between workspaces, were provided through two elements of the Shared Stage Window: Synchronised Stage View and Data Structure Diagram. These allow unique collaborative 3D interaction during a collaborative session. A user scenario was also illustrated to show how the Syco3D system could be used for the collaborative construction of a vehicle model. The details of software structure were also described. In the Syco3D system, the procedures and event handling mechanisms replicate and synchronise the database and representation in the Shared Stage Window. Through the evolution of the prototype, many ways of achieving a Shared Stage system were investigated and it was demonstrated that the Shared Stage framework was feasible for building a real-time collaborative application for sophisticated 3D CAD tasks.

Chapter 6

Usability Evaluation



6.1 Introduction

In previous chapters, the Shared Stage has been suggested as a shared workspace for real-time collaborative 3D CAD work and ways of investigating a prototype system based upon the framework have been explored. As a result, the prototype system of a real-time collaborative 3D CAD system, Syco3D, was built and the Shared Stage module was incorporated. The next step of the research was to examine the feasibility and the impact of the new tool in real-time collaborative 3D CAD activities through usability evaluation.

The evaluation was carried out as a laboratory experiment. For comparative evaluation, a single user version of the Syco3D system was also developed. Team designers accomplished experimental 3D modelling tasks using two different computer based 3D modelling tools: a conventional single user 3D modelling tool using file exchange processes and the real-time collaborative 3D modelling tool incorporating the Shared Stage module. The outcome from the group task, verbal interactions, and user satisfaction were evaluated quantitatively, while several aspects relating to the collaborative process were evaluated qualitatively. In the following sections, the objectives, methods and findings of the usability evaluation are illustrated.

6.2 Objectives

One main objective of the usability evaluation was to examine the impact of the Shared Stage and the usefulness and usability of the new system. The impact of the Shared Stage on real-time collaborative 3D design activities can be investigated through answering the following questions:

i) Does the addition of the Shared Stage help participants produce a better 3D model?

ii) Does it provide a positive or negative impact on the collaborative process?

iii) Do participants show the preference on the addition of the Shared Stage for real-time collaborative 3D design tasks?

Questions to examine the usefulness and usability of the new system may include:

i) Do designers make good use of the collaborative system provided?,

ii) Do designers use the system efficiently?

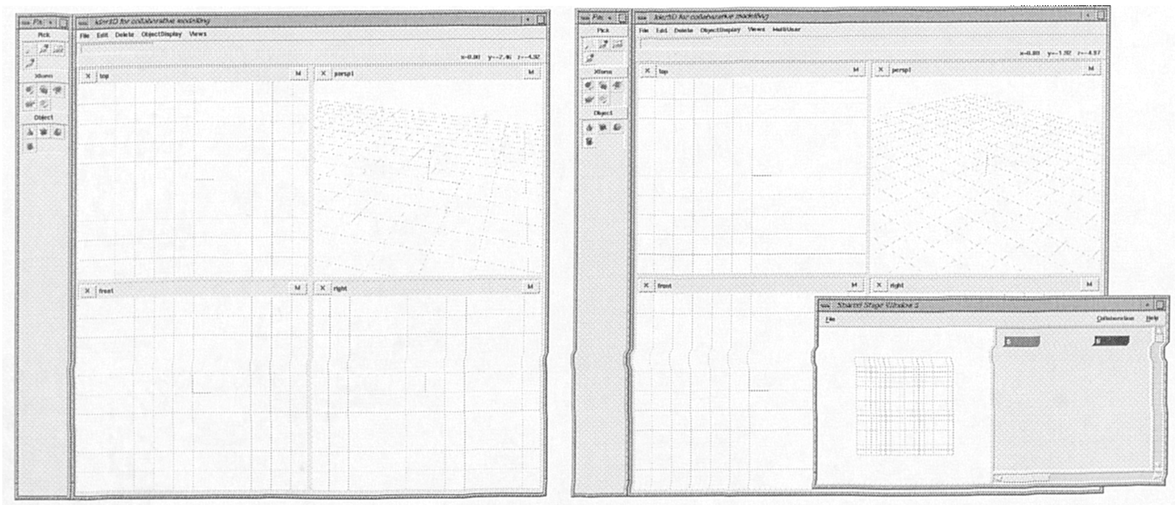
iii) Do designers use the Shared Stage frequently?

The evaluation also aimed at identifying the impact of the Shared Stage on various 3D CAD activities that might take place in different phases of the design process. Because designers engage in various 3D design tasks in different phases of the design process, the impact and the usefulness of the Shared Stage may be different. An early phase of the design process was of interest in the evaluation, since users focus on conceptual 3D design work involving frequent changes to the 3D properties of the model, such as dimension and proportion. The detail design phase was also of interest as many of the model specifications are then finally decided and created according to the agreed specification.

6.3 Method

6.3.1 Systems used in the experiment

To conduct a comparative usability evaluation, a single user version of the Syco3D system was developed. The single user version was simply implemented by removing procedures and menu items related to the Shared Stage. Although the single user version does not have the Shared Stage Window and collaborative features, it has the same user interface, functions and layout. Therefore, the user interface of a single user version was almost identical to the original version of the Syco3D system. In the experimental session, with the original real-time collaborative version, the Shared Stage module was used as a shared 3D workspace for group work, whilst in a session with the single user version it was necessary to exchange 3D data files. Figure 6.1 shows the screen layout of two systems used in the experiment.



The single-user version without the Shared Stage module — The multi-user version with the Shared Stage module

Figure 6.1 Screen layouts of the single user and the multi-user version of the Syco3D system

6.3.2 Participants

Eight pairs of students were recruited from the department of Design at Brunel University. Most had basic skills of interpreting 3D shapes from technical drawings, and were experienced in the use of basic 2D and 3D graphic applications. Therefore, it was expected that they could accomplish the experimental 3D design tasks without learning the basic functions of 3D CAD tools. In addition, because of the similarity of basic geometric functions and the user interface between the prototype and standard 2D and 3D graphic applications, all participants could quickly learn how to use the experimental systems.

6.3.3 Experimental setup

Two Silicon Graphics workstations were connected via LAN and placed opposite each other as shown in Figure 6.2. Participants sitting opposite each other could talk to and see each other, but they could not see each other's computer screen. This layout is a common arrangement between collaborating people in a co-located situation. It can also be extended for a distributed situation using interpersonal communication systems, such as Clearboard (Ishii et al., 1995), or VideoDraw (Tang and Minneman, 1991).

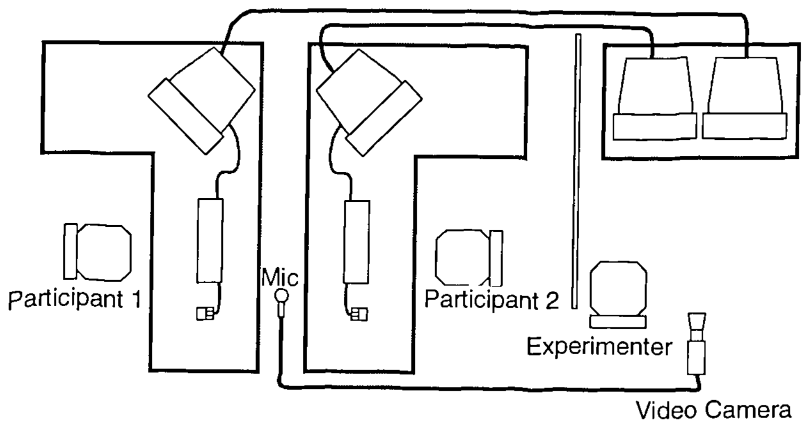


Figure 6.2 The layout of the experiment room

6.3.4 Tasks

In order to examine how collaborative tools are used in different phases of the design process, two modelling tasks were designed. The first task was to build a 3D desk and chair from a conceptual drawing. The drawing showed all the geometric forms in orthographic and perspective views but did not include any dimensions. This type of 3D modelling task is common in a conceptual design phase, when designers need to quickly visualise their ideas in 3D form without dimensional accuracy. Figure 6.3 (top) shows the drawing that was given to the participants for the first task. The second task was to build a 3D model of a telephone charger from a dimensioned drawing. This type of task is accomplished in the detail design phase, when major decisions about the concept and specifications of detail are decided. As all the dimensions and the specifications were fixed, participants had to focus on building the model rather than making decisions about the shape or the configuration. Figure 6.3 (bottom) shows the dimensioned drawing given to participants. A solid modelling application, Pro-Engineer, was used to produce the two drawings. For both tasks, it was considered that the experimental tasks were moderately complex requiring group collaboration, but appropriate for completion within experimental conditions using the prototype systems.

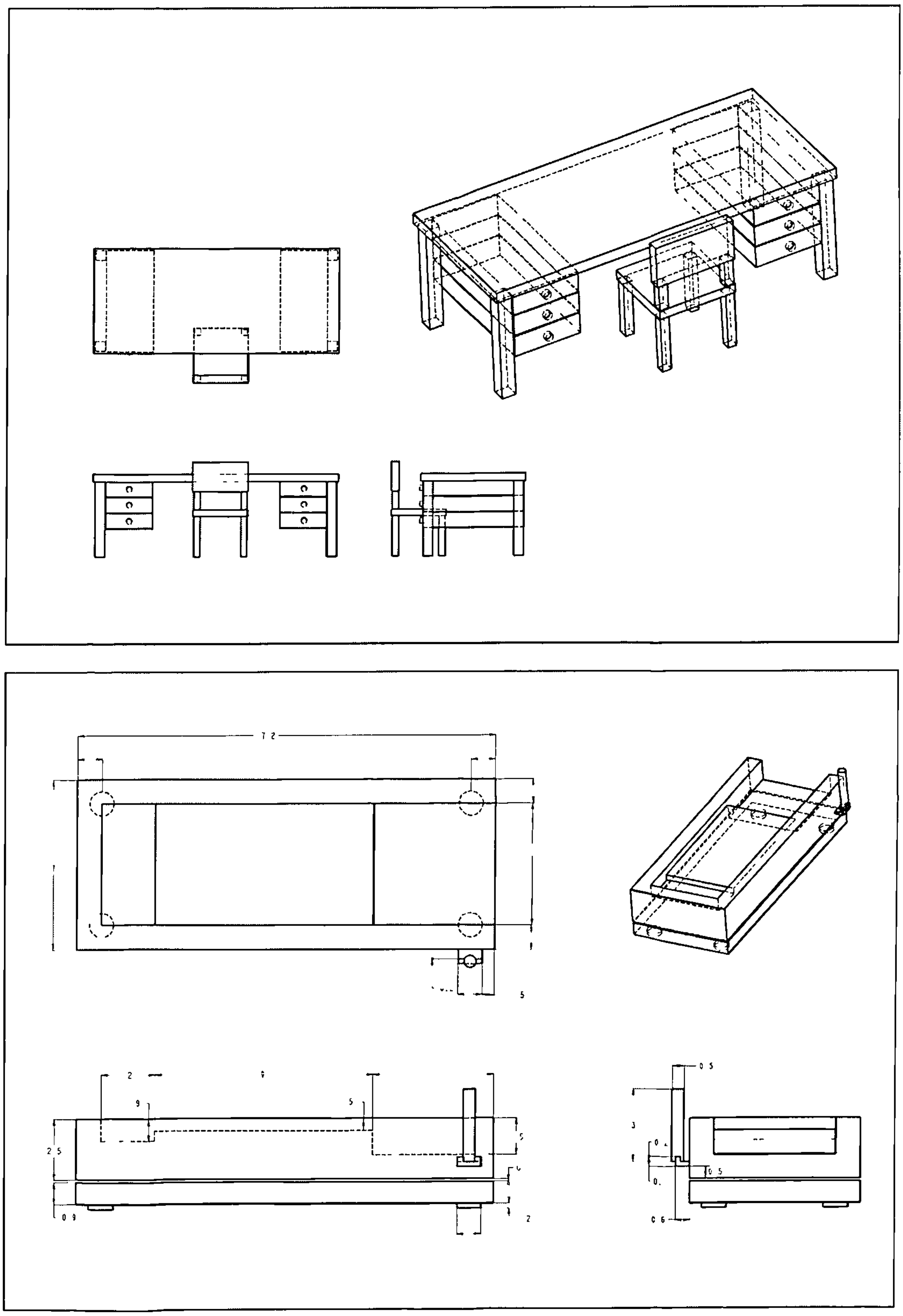


Figure 6.3 The experimentation tasks. The first task of building a 3D CAD model of a table and a chair from a conceptual 3D drawing is shown above and the second task of building a 3D CAD model of a telephone charger from a dimensioned drawing below.

6.3.5 Procedures

Participants were given an overview and demonstration of the versions of the Syco3D system and carried out two simple modelling tasks both individually and collaboratively in order to become familiar with the system. The tasks had many aspects in common with the experiment tasks, in terms of the way parts are created. Participants should therefore have been given sufficient experience with the interface of the system and its use to build 3D models.

After the exercise tasks, they were asked to perform the actual experiment. They performed each task in two conditions. In one condition they worked together to carry out the experimental task, exchanging files using the single user version (Condition NS), without the Shared Stage Window, while in the other condition they used the Shared Stage Window of the Syco3D system for collaboration (Condition S). In order to minimise the effect of task order in the results, the sequence of the experiment tasks and the use of the system was counterbalanced as shown in Table 6.1. Independent sampling between groups was used to compare the two conditions. In addition, the user preferences and satisfaction between these conditions were compared within a group. The questionnaire about their preference and satisfaction with the system used were completed when they finished each task.

Table 6.1 The sequence of experiment conditions

	Task 1	Task 2
Pair 1	Condition NS	Condition S
Pair 2	Condition S	Condition NS
Pair 3	Condition NS	Condition S
Pair 4	Condition S	Condition NS
Pair 5	Condition NS	Condition S
Pair 6	Condition S	Condition NS
Pair 7	Condition NS	Condition S
Pair 8	Condition S	Condition NS

Condition S: Using the Shared Stage Module,
Condition NS: Using File Exchange (without the Shared Stage Module)

6.3.6 Data collection and analysis methods

Data collected in the experiment included experimenter observation, video recordings of the computer screens with dialogues, questionnaires filled out by the participants after each experiment task, and recordings of interviews. The collected data was used for both quantitative and exploratory analysis. The purpose of the quantitative measures was to determine if there was significant interaction between the tools used and the actual design and process of collaboration. Quantitative measures focused on the outcome of the group task, the collaborative process and user perception about the tools (Table 6.2).

Table 6.2 Quantitative measure

Type of measure	Method
Outcome	- By checking the time to complete the experiment tasks
Process (verbal Interaction)	- By counting dialogue segments, conversation turns and content categories - By measuring the time spent in verbal communication
User conception	- By measuring the scale value of user response on the satisfaction of the outcome and the collaborative process

Exploratory analysis focused on the collaboration process and the usability of the systems. In particular, the strategies used by participants between each condition were analysed and compared. Usability and interface issues of the system were also a main concern in the exploratory analysis. The group interactions not detected by quantitative methods were investigated through reviewing transcribed dialogues.

In the following sections, the result of quantitative measures are first reported and the exploratory analysis is then presented in terms of the strategies of group approach and the particular group interaction of interest. The role and user interface issues of the Shared Stage module observed are also discussed by presenting example interactions.

6.4 Quantitative results and analysis

6.4.1 Completion time for the experimental tasks

The outcome of group work can be compared by assessing the quality of the resulting 3D model and the time to complete the task. Quality of the finished 3D model however is a subjective issue, as the assessment of the quality may be different and sometimes general agreement can be difficult to achieve among judges. Therefore, the focus for the outcome measure was on the time for completion of the experimental tasks.

Although the completion time for the actual experimental task was of importance for comparison, the time spent in demonstration and the practice exercises with the tool also provides an indication of how quickly and easily participants became familiar with the system. The average time for introduction and demonstration of the system was 40 minutes. None of the pairs were exceptionally fast or slow in the practice exercises. The participants took about 30 minutes to complete the exercise tasks in condition NS using the single user version of the system, while they finished within 15 minutes in condition S with the multi-user version. The participants became comfortable using the functions and interface of both systems during the introduction and the exercise period, which took about one and half hours in total.

As each task was accomplished in two different conditions, there were four cases to measure; the completion times of Task 1 in condition NS, Task 1 in condition S, Task 2 in condition NS and Task 2 in condition S. The completion times for these cases are summarised in Table 6.3, and shown in Figure 6.4. The average time to complete task 1 in condition NS was 37 minutes, while it took 29 minutes in condition S. The average time to complete task 2 was 29 minutes in condition S and 35 minutes in condition NS. The average completion time of condition S was shorter than condition NS in the first task in which the group has to decide upon dimensions and relative scale of each part of the model. Whereas, in the second task, when all the dimensions and scales were given, the average completion time of condition S was slightly longer than condition NS.

Table 6.3 Summary of completion times for experimental tasks in both conditions (minutes)

Task	Condition	Maximum	Minimum	Mean	δ
1	S	28.83	25.50	26.95	1.44
	NS	37.17	26.68	31.43	5.36
2	S	40.17	27.00	34.85	5.60
	NS	38.43	18.17	29.83	9.78

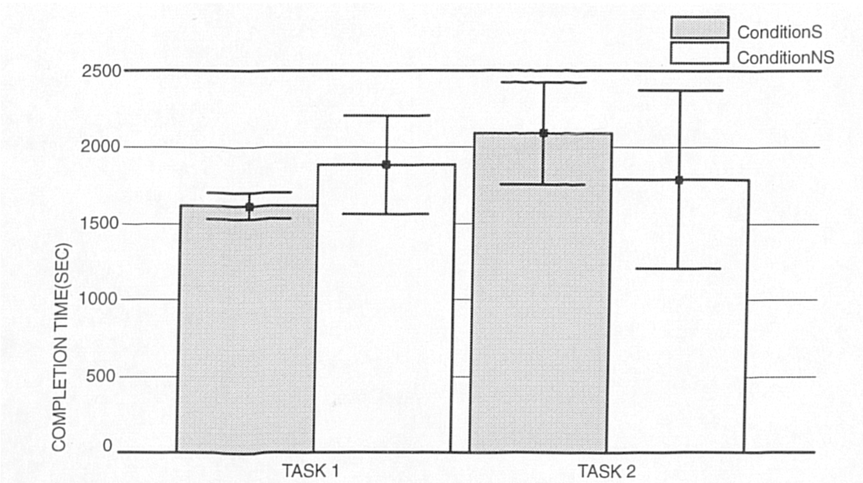


Figure 6.4 Comparison of the average and standard deviation of completion time

The relationship between the independent variable task and condition has been compared using the Mann-Whitney U test. The statistical interaction between task and condition was not detected (U=4.0, W=14.0, Corrected for ties 2-Tailed P=.2482), although the figure shows a slight difference in means.

Groups who were using the Shared Stage module were expected to finish the tasks quicker than the groups using the conventional file exchange method. However, it appeared that the direct influence of the Shared Stage module was less visible in the measure of completion time. This result may be explained by the fact that the impact of the system may not be detectable if the effect is not big enough in group work (McCarthy et al., 1991; McCarthy and Monk, 1994). People in a group tend to sacrifice any secondary objectives to accomplish the major objectives of the task. The different results between the first and the second task indicate that the Shared Stage may be used differently in different phases.

6.4.2 Verbal interaction analysis

Verbal interactions were measured to examine and compare the collaborative processes between conditions. In order to analyse the interactions quantitatively, all the dialogues in the experimental sessions were transcribed with a time code and the interaction with the system was captured with a video recording of the computer screen. A method of quantifying verbal interactions was developed using dialogue segments. A dialogue segment was defined as a block of continuous conversation in a given time. It was segregated when there was a significant gap between them. Longer and frequent dialogue segments may indicate more active verbal interaction between participants. The focus was on the number and the occurrence pattern of dialogue segments during the sessions, the number of conversation turns and the time spent for verbal communication. Similar measures have been used by researchers to show the pattern and efficiency of collaborative processes (Olson et al., 1992; Gutwin, 1997; Gutwin and Greenberg, 1998; Goldschmidt, 1995). The measure of these variables was also used as background information for the exploratory analysis that will be discussed in a later section. The number of turns and the time spent in verbal communication per dialogue segment were calculated from the initial results, as well as the proportion of time spent for verbal communication in a whole session.

In order to identify the kinds of collaborative interactions used to accomplish the 3D tasks and the impact of the new system, the context of the verbal interactions was also analysed. Verbal interactions were categorised through an extensive review of the transcription and measured the distribution and occurrence of each interaction category. The frequency ratio of each interaction category to the total number of dialogue segments was also calculated.

6.4.2.1 Dialogue segment measures

The number of dialogue segments can be used to investigate the level of interaction for each pair. An initial expectation was that more dialogue segments would be found in condition S than condition NS regardless of the type of task. The result of the numbers of dialogue segments in all conditions is summarised in Table 6.4 and the main descriptives are shown in Figure 6.5.

Table 6.4 Summary of the number of dialogue segments

Task	Condition	Maximum	Minimum	Mean	δ
1	S	22.00	15.00	18.00	2.94
	NS	21.00	6.00	15.25	6.90
2	S	31.00	13.00	22.00	8.41
	NS	26.00	8.00	16.75	7.63

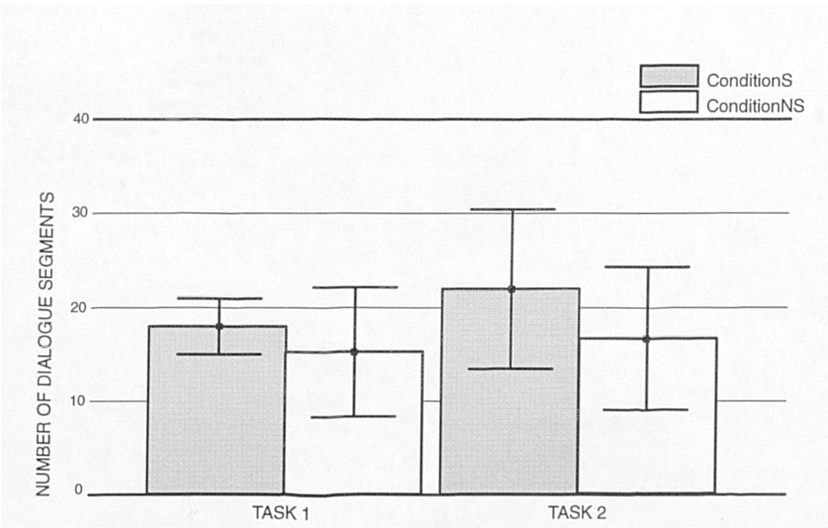


Figure 6.5 Comparison of the average number of dialogue segments

The pairs had between 15 to 22 dialogue segments whilst undertaking Task 1 in Condition S, and between 6 and 21 in condition NS. The mean number of dialogue segments in Task 1 was measured as 15 for condition S and 18 for Condition NS. Although the figure shows some difference, the statistical test using Mann-Whitney U test did not show any significance in the interaction between condition and task 1.(2-Tailed P=.5637)

In Task 2, it was noted that participants had more dialogue segments in condition S than in condition NS. The average number of dialogue segments in Condition S was measured as 22, while it was 16.75 in Condition NS. The numbers ranged between 8 and 26. It was also noted that in Condition NS participants had a long silent period in the middle part of the collaborative session. The analysis of the relationship between the independent variable task and dependent variable condition again fails to show any significance of interaction between task and condition (2-tailed P=.3865).

Although the test did not show significant interaction between tasks and conditions, the distribution of the dialogue segments indicates the difference between the conditions. The distribution of the occurrence of verbal interaction is shown in Figure 6.6 (Top). The grey bars indicate the periods when participants were engaged in dialogue. As can be seen in the figure, dialogue segments for task 1 in condition S occur frequently, so that there is little time where there are no interactions for more than 5 minutes. This indicates that there were continuous mutual interactions throughout the collaborative session. On the other hand, for the same task in condition NS, periods where there is no interaction can be found in many places in the figure. The results were similar in the second task. The verbal interactions in condition S were more widely distributed throughout the session, while in condition NS the 'no-verbal' interaction period tended to occur more often and last longer. This result indicates that the Shared Stage module motivated more verbal interactions and helped close collaboration amongst users.

Figure 6.6 The distribution of verbal interactions. Task 1 in condition S and condition NS (Top). Task 2 in condition S and condition NS(Bottom). Grey bar indicates the period when participants had verbal interaction.

6.4.2.2 Turns

The number of conversation turns provides information about how often the interchange of information took place during the experiment session. The larger number of turns represents more active collaboration during the session. Table 6.5 shows the summary of the number of turns in task 1 and task 2. Figure 6.7 represents the mean comparison.

Table 6.5 Summary of the number turns

Task	Condition	Maximum	Minimum	Mean	δ
1	S	191.00	82.00	124.50	50.40
	NS	199.00	50.00	110.25	63.44
2	S	165.00	101.00	132.50	30.39
	NS	198.00	44.00	95.25	69.64

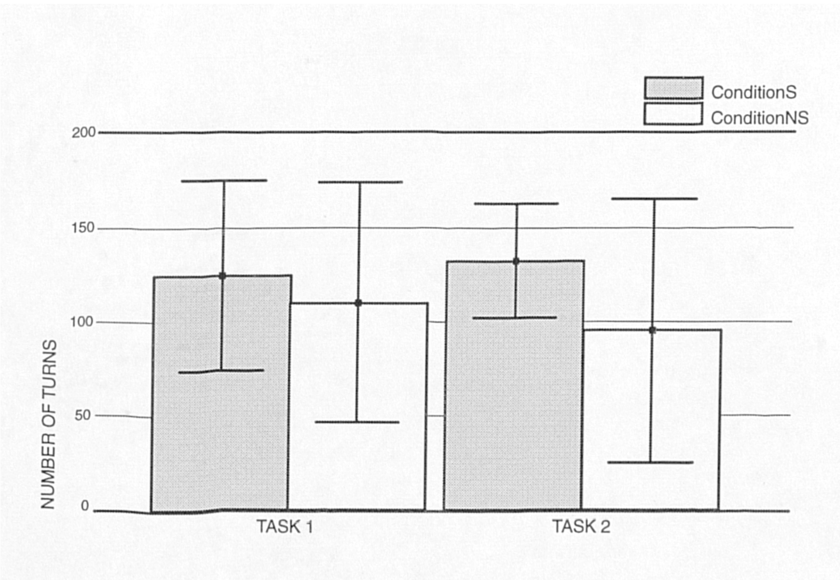


Figure 6.7 Comparison of the average number of turns

The average turns for task 1 was 124.5 in condition S and 110.25 in Condition NS. It ranged from 82 to 191 in condition S and 50 to 199 in Condition NS and a larger variance between groups in Condition NS was measured. In task 2, the average number of turn taking was measured as 132.5 in condition S and 95.25 in Condition NS. Although there was a greater difference in mean values, a larger variance was found in conditions of task 2. It ranged from 101 to 165 in condition S and 44 to 198 in Condition NS. There was no statistical interaction between task and condition with regard to the number of turns (Mann-Whitney U test, 2-Tailed P=.7728 in Task 1, P=.2482 in Task 2).

The average number of turns in a dialogue segment for task 1 was 7.14 in condition S, and 8.2 in Condition NS. For task 2, it was 6.35 in condition S and 5.84 in Condition NS. The summary of the number of turns per dialogue segment is presented in Table 6.6 and shown in Figure 6.8.

Table 6.6 Summary of the number of turns per unit dialogue segment

Task	Condition	Maximum	Minimum	Mean	δ
1	S	10.61	4.05	7.14	3.20
	NS	14.50	3.57	8.20	4.88
2	S	7.77	4.87	6.35	1.20
	NS	8.13	2.32	5.84	2.65

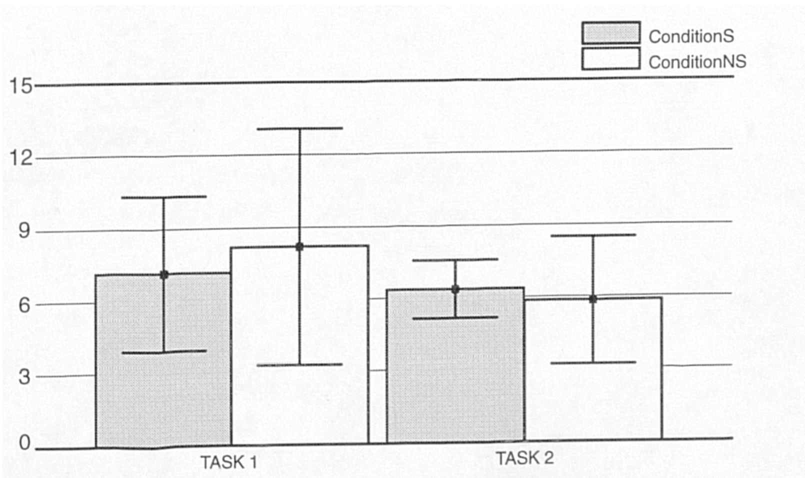


Figure 6.8 Comparison of the average number of turns per unit dialogue segment

There was also no statistical interaction found between task and condition with regard to the number of turns (Mann-Whitney U test, 2-Tailed P=.7728 in Task 1, P=.2482 in Task 2). The average number of turns in a unit dialogue segment did not reveal a difference between the two conditions for either task. With regard to the average number of turns in a dialogue segment, it can be inferred that the verbal interaction of the participants had a similar pattern in each dialogue segment regardless of the tasks and conditions. The total number of turns shows the difference between conditions for both tasks. It can be suggested that the Shared Stage module encouraged more conversational exchange and turn taking.

6.4.2.3 Time of verbal interaction

The time length of verbal interaction also indicates the interactivity of collaboration processes. The results for the length of verbal interaction are summarised in Table 6.7, and the means are shown in Figure 6.9. The ratio of the time spent for verbal interaction to the total completion time was calculated and is represented in Table 6.8, and Figure 6.10.

Table 6.7 Summary of the time length of verbal interaction (second)

Task	Condition	Maximum	Minimum	Mean	δ
1	S	970	540	712	183
	NS	961	225	600	329
2	S	1357	366	821	422
	NS	845	328	549	220

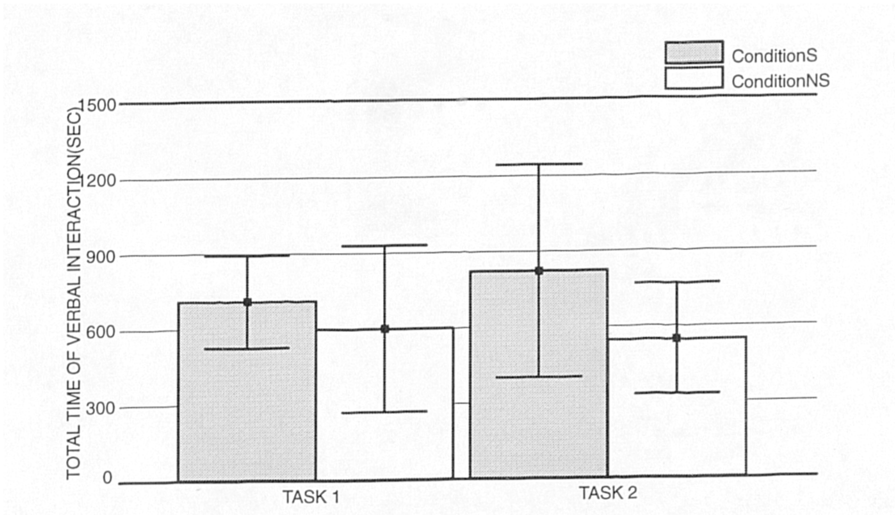


Figure 6.9 Comparison of the average time length of verbal interaction

Table 6.8 The ratio between time length of verbal interaction and total completion time (percent)

Task	Condition	Maximum	Minimum	Mean	δ
1	S	61.59	35.39	44.14	11.89
	NS	45.98	14.05	30.48	13.37
2	S	41.37	22.59	33.56	8.75
	NS	51.56	20.46	32.55	14.67

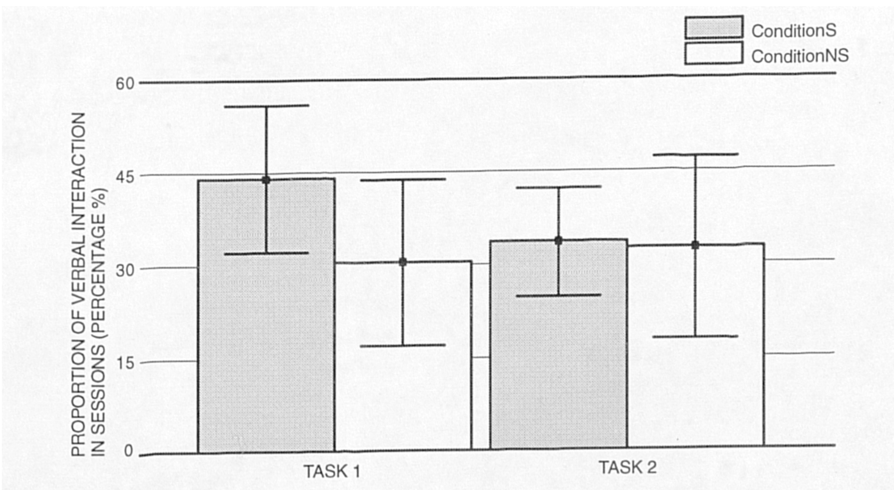


Figure 6.10 Comparison of average ratios between time length of verbal interaction and total completion time

As shown in Table 6.7, during task 1 participants took approximately 9 to 16 minutes for each dialogue segment in condition S and between 3.75 and 16 minutes in Condition NS. Table 6.8 shows that the ratio of time spent for verbal interaction in condition S(44%) scored larger than the ratio in condition NS(30%). For almost half the session participants were engaged in verbal interaction in condition S. On the other hand, for task 2 participants took between 6.1 and 22.62 minutes in condition S, and between 5.47 and 14 minutes in Condition NS. A similar ratio of the time for verbal interaction to the total completion time was seen for both conditions. About 33 percent of the time was spent for verbal interaction in condition S and 32 percent in Condition NS.

The time spent in each dialogue segment was calculated and summarised in Table 6.9 and Figure 6.10. As seen from the table, each dialogue segment took about 40 seconds and the average length of verbal interaction time per dialogue segment is similar in both conditions. The overall results indicate that in condition S participants spent more time in verbal interaction. It can be inferred that the Shared Stage module causes the collaborative process to involve more verbal interaction.

Table 6.9 The time spent in each dialogue segment

Task	Condition	Maximum	Minimum	Mean	δ
1	S	64.70	30.80	40.99	16.02
	NS	74.00	16.10	43.60	23.88
2	S	49.80	21.50	37.25	12.38
	NS	70.30	17.30	38.23	22.57

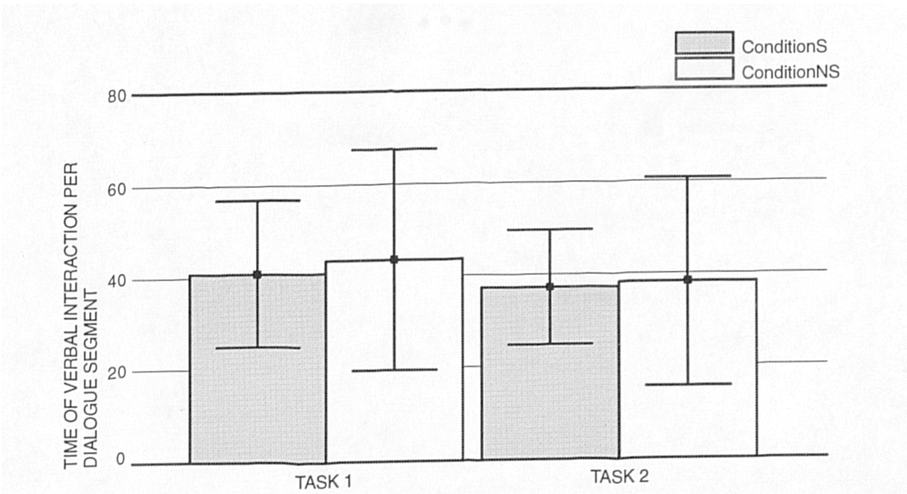


Figure 6.11 Comparison of the average time spent in each dialogue segment

6.4.2.4 Context analysis

Verbal interactions were also analysed in terms of the context of the group interactions. The activity classification was developed and the occurrence of each category in dialogue segments was measured. The classification was evolved through extensive review of the contents of the verbal interaction and the literature review on the studies of group work. According to previous researches, group work typically falls into two broad classes: task-related activity and executive activity (Olson et al., 1992; Cross et al., 1996). The first refers to work on the task itself, for example the 3D modelling activity and the second to those activities used by groups to manage their interactions, including co-ordinating their tasks, deciding what to do next and reporting on each others progress. These broad classifications were used and they were subdivided into seven sub-categories depending on the focus of the contexts. Descriptions of each activity are as follows.

a) Task-related interaction

In the class of task-related interactions, four sub categories were identified.

- Model-centred interactions: These are focused on 3D objects under construction. An example activity is to decide the dimension, scale and the location of parts of the model.
- Method-centred interactions: These are focused on the techniques or methods that are applied to build the models. This is related to the ways to the process of creating components and their combination.
- Interactions related to shared outcome: These are centred on the shared outcome that is produced during the collaborative sessions.
- Interactions related to system functions: This category is related to the activity of solving problems or functions of the systems. They discussed the detailed functions of the system and consulted each other about ways to solve problems that they encountered.

b) Executive interaction

Interactions in this class have been divided into three types.

- Planning process: This concerned planning the collaborative process. For example, including how to assign people to certain activities.
- Co-ordination of process: This is related to such co-ordination activities as monitoring and reporting on on-going activities.
- Digression and others: These are activities not directly related to the main objective of the session. Examples are people joking, discussion of side topics and interruptions not related to the session.

None of these activities are exclusive, so a dialogue segment may have multiple interaction categories. Once the classifications were defined, a sample session (Appendix B.1) was coded by three researchers to establish the reliability of the categories. As the similarity was found between the coding results, the rest of the sessions were coded by the author.

After classification, the occurrence of each category was measured in each session. In addition the ratio of the number of dialogue segments where interaction categories occurred and the total number of dialogue segments were calculated. The frequency of these interaction categories and the ratio are summarised in Table 6.10 and shown in Figure 6.12. Firstly, in condition S, the most frequent was that concerned with the shared outcome. Half of the dialogue segments in condition S included the occurrence of interaction focused on the shared outcome. The average ratio of occurrence was scored as 48.9%. The second most frequent interactions centred on modelling methods, scoring 44.0%. This indicates that participants frequently consulted each other. On the other hand, in condition NS, checking progress was the most common activity scoring 42%. The frequency of other interaction categories, including planning and digression activities, were similarly distributed. The frequencies of all interactions under condition S is much higher than those of condition F. Particularly, that focused on the shared outcome shows the greatest difference between the two conditions. Since the Shared Stage module provided consistent feedback of the outcome, participants had more chances to initiate this interaction. Under Condition NS, process co-ordination scored highly. In many cases it related to the interaction of making inquiries about the participants' progress. The result explains the lack of awareness support under Condition NS. Based upon the results from interaction categories, it is inferred that the participants could interact in more sophisticated and co-operative interactions under condition S, because the Shared Stage module provided basic awareness of their participants' progress.

Table 6.10 The average number of dialogue segments in which each interaction category took place, and its proportion to the total number of dialogue segments

Average		C1	C2	C3	C4	C5	C6	C7	All sessions
Frequency	Cond. S	2.63	4.63	8.5	6.9	5.6	9.9	0.8	20
	Cond.NS	2.4	4	5.3	7	4	5.1	0.8	16
Proportion	Cond. S	13.5%	24.5%	44.0%	34.8%	29.4%	48.9%	3.6%	100%
	Cond.NS	14.6%	26.0%	37.6%	42.0%	28.0%	34.0%	5.5%	100%

C1: Planning processes
C2: Model-centred interactions
C3: Method-centred interactions
C4: Co-ordination of processes
C5: Interactions related to system functions
C6: Interactions related to the shared outcome
C7: Digression and others

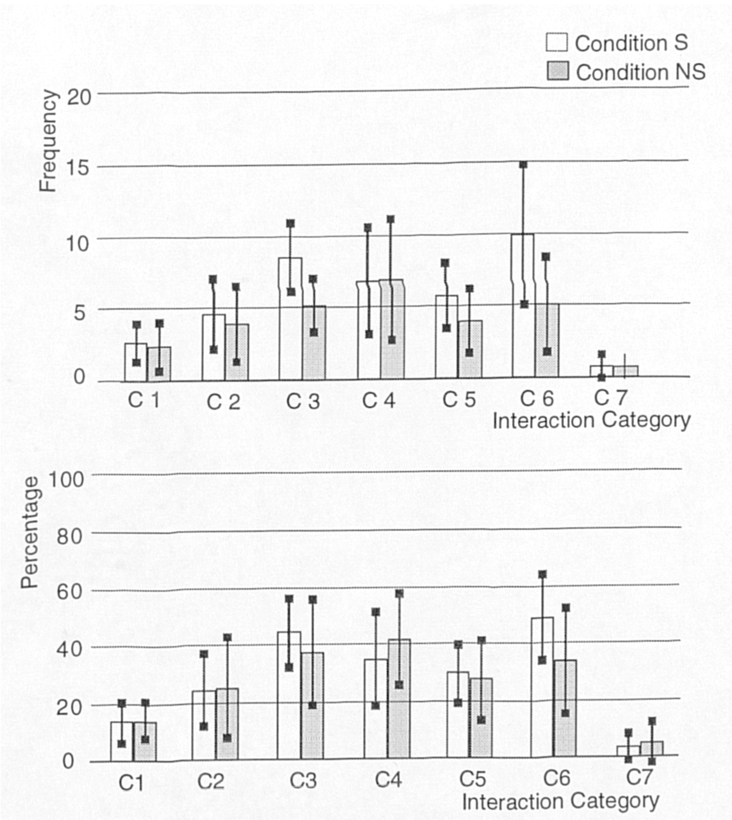


Figure 6.12 Comparison of the occurrence(top) and the proportion(bottom) of each interaction category. The proportion is the ratio of the number of dialogue segments, in which an interaction category took place, to the total number of dialogue segments (bottom) for each condition.

6.4.2.5 Patterns of use of the Shared Stage Window

The pattern of accessing the Shared Stage Window shows the usefulness of the Shared Stage module during the collaborative session. The Shared Stage module was frequently used throughout most sessions under condition S. Table 6.11 shows the ratio of the number of dialogue segments which discussed access to the Shared Stage module to the total number of dialogue segments in each session. The ratio ranged between 25.9% and 76.5% and the average value was 39.2%. The access ratio indicates that the Shared Stage Window is closely related to the dialogue segment. It can also be said that during collaborative interaction, participants made good use of the Shared Stage module. This result indicates the close relations between access to the Shared Stage module and verbal interaction among group members.

Table 6.11 Number of Dialogue Segments with the use of the Shared Stage module

Pair ID	Number of Dialogue Segments with the use of the Shared Stage module (Percentage)	Total Dialogue Segment
P1	7 (46.7%)	15
P2	2 (15.4%)	13
P3	9 (50.0%)	18
P4	9 (29.0%)	31
P5	9 (40.9%)	22
P6	5 (29.4%)	17
P7	13 (76.5%)	17
P8	7 (25.9%)	27
Average	7.6 (39.2%)	20

6.4.3 Participants response measure

6.4.3.1 Perception of the outcome and collaboration process

The user perception of the outcome and the process were also measured in the questionnaire, after each task. Questions involved four aspects of user perception to the collaborative 3D design task. They were: i) how satisfied they were with what they produced, ii) how satisfactory the collaboration process was, iii) how effectively they were able to collaborate, and iv) how well they knew their partners activities during the collaborative session? (Appendix B). The question used a five point scale ranging between positive and negative. The responses were translated to interval scores, using 5 to represent the most negative and 1 to present the most positive. Table 6.12 summarises the mean responses for each question, and Figure 6.13 shows the mean values in a bar chart.

Table 6.12 Summary of questionnaire response of satisfaction

Question 1 (Q1): How satisfied were you with what you produced?

Question 2 (Q2): How satisfactory was the collaboration process?

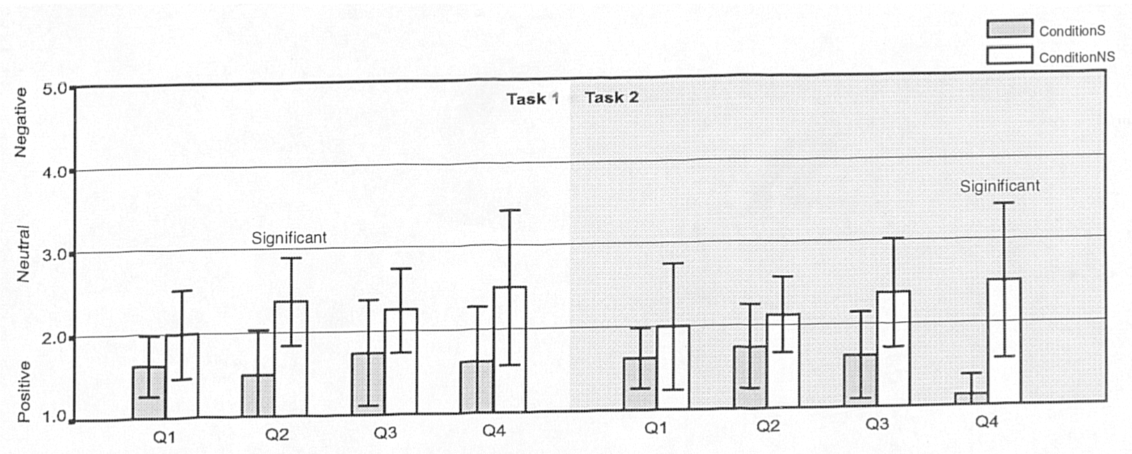
Question 3 (Q3): How effectively were you able to collaborate?

Question 4 (Q4): How well did you know your partners activities during the session?

Task	Condition	Q1		Q2		Q3		Q4	
		Mean	δ	Mean	δ	Mean	δ	Mean	δ
1	S	1.63	0.52	1.50	0.76	1.75	0.89	1.63	0.92
	NS	2.00	0.76	2.38	0.74	2.25	0.71	2.50	1.31
2	S	1.63	0.52	1.75	0.70	1.63	0.74	1.13	0.35
	NS	2.00	1.07	2.13	0.64	2.38	0.92	2.50	1.31

5: most negative, 1: most positive

As it can be seen in the table, participants expressed high satisfaction about the outcome and collaborative process in condition S. The efficiency and awareness support were also scored higher in conditions S. The responses from condition S and Condition NS on each question were compared using Mann-Whitney U test. The responses of Question 2 in Task 1 and Question 4 in Task 2 showed significant differences (2 Tailed $P < 0.05$). The user response indicates that the Shared Stage Window significantly affects the participants' perception of the collaboration process and awareness support.



Q1 : Outcome (I am Satisfied with what we produced)
Q2 : Collaboration Process (I am satidfied with the collaboration process)
Q3 : Efficiency (We were able to collaborate effectively)
Q4 : Awareness (I knew what my partner was working on during the collaborative session)

Figure 6.13 Comparison of questionnaire responses

6.4.3.2 Usability and efficiency

Participants were also asked about the usability and efficiency of the Shared Stage Window. Their responses were translated to interval scores, using 1 to represent most positive and 5 to represent most negative. Table 6.13 shows the questions about the user interface and system efficiency, from the questionnaire, and the mean value of the responses.

The participants reported that although the system had been implemented in a prototype level, it could be used for realistic collaborative modelling activities. General 3D interface issues such as navigation in the 3D modelling view appeared easy to use (mean response: 2.06). In terms of the interface of the Shared Stage Window, participants also found it easy to use (mean response: 1.94) and the Synchronised Stage view was valuable to group work (mean response: 2.0). Locking mechanisms did not interrupt the use of the synchronised camera (mean response: 3.38). They were greatly interested by the telepointer, during the practice exercises, however during actual construction tasks it was not often used (mean response: 2.69). The role of the telepointer changed between occasions when the participant's aim was to modify existing models and when the aim was largely for communication activities. Awareness support that allowed other participants to know the state of the shared workspace appeared to be useful for group work. For example participants reported that the red outline indicating who had the control of Synchronous Stage view, and the little rectangle symbol under object nodes indicating reference state, were useful because they gave information about partner's state in the Shared Stage window (mean response: 1.62). Participants found it convenient to use the functions of exchanging and referencing objects between individual workspaces by connecting object nodes with user nodes in the Shared Stage Window (mean response: 1.88). The user interface of data exchange and referencing was easily understood and used by participants (mean response: 1.62).

Table 6.13 Results fom the questions about user interface and system efficiency

Question	Mean*	δ
The system speed interrupted the 3D modelling process.	3.25	.77
Navigation was easy to use.	2.06	1.12
The Synchronised Stage view was easy to use.	1.94	.85
The synchronised camera was valuable in performing the task.	2.00	.82
Locking mechanisms for the control of the shared stage view caused a conflict.	3.38	.81
The telepointer played an important role in communicating with my partner.	2.69	1.20
The red outline indicating who has control provided useful information about the state.	1.94	.68
Referencing and transporting provided a valuable function for collaborative modelling.	1.62	.62
The interface of referencing by object node was easy to understand and use.	1.88	.89
Rectangle for reference state provided valuable information.	2.00	.82
The size of shared stage window was too small.	3.06	1.12
A bigger shared stage window would be better, despite hiding part of the modelling views	3.43	1.09
The shared stage window improved the quality of output	2.43	.85
The user interface of the shared stage window was difficult	4.00	.55
Allowed to perform parallel tasking.	1.31	.48
Allowed to allocate the work effectively	1.50	.82
Allowed efficient exchange of object data	1.62	.62
Allowed integration of individual output effectively	1.44	.51

* 1 represents most positive and 5 most negative

In terms of the size of the Shared Stage window, although it was a valuable addition to the 3D modelling system, they would have preferred larger individual workspace views. Several participants wanted to change the size of the window dynamically whenever they wanted to interact with the shared workspace. Although participants preferred keeping the Shared Stage window during the task, they were unsure about its direct contribution to the quality of output. It can be inferred that the direct impact of the Shared Stage window might not be easily visible on the output, as discussed in the outcome measure. In the user response measure, participants responded very positively to most of the questions about the usability and efficiency of the Shared Stage module. This indicates that the Shared Stage window improved the efficiency of the collaborative modelling process, helping teams to perform parallel tasking and facilitating efficient data exchange.

6.4.3.3 Preferences

Finally, participants were asked three questions about their preference in condition. The questions include: i) which condition better supported collaborative tasks?, ii) which condition was easier to use for group tasks?, and iii) which system did participants prefer overall? Almost all participants responded with a preference towards condition S, as shown in Table 6.14.

Table 6.14 Results of user preference

Question (Which condition ..)	Condition S	Condition NS
1. ... better supported collaborative modelling tasks	14	2
2. ... was easier to use for group work	12	4
3. .... do you prefer	14	2

These responses were analysed using a one-way Chi-Square test, summarised in Table 6.15. Alpha was maintained at 0.05. The number of participants choosing the condition S was significantly higher than the expected number for question 1 and 3.

Table 6.15 One-way Chi-Square test of preference

Questions (Which condition .)	Chi-Square	D.F.	Significance
1. ... better supported collaborative modelling tasks	9	1	0.0027 ($p<0.05$)
2. ... was easier to use for group work	4	1	0.0455 ($p<0.05$)
3. .... do you prefer	12.5	2	0.0019 ($p<0.05$)

Fourteen out of sixteen responses preferred the Shared Stage condition to the condition using file exchange. Condition S was preferred for both supporting group work (14 out of 16) and easier to use (12 out of 16). Some participants commented that it can be distracting to view the activities of colleagues when concentrating on an individual task, although it provides an easy way to exchange the output of individual tasks. The result demonstrates that the Shared Stage module is a valuable addition for the collaborative 3D modelling task.

6.4.4 Summary of the quantitative results

The completion time results showed that the average value under the condition with the Shared Stage module was shorter in the first task but longer in the second task. This implies that the Shared Stage module may be more effectively used in a conceptual design phase and the level of usefulness of the module varies with the design process.

The results of verbal interaction showed that there were differences in terms of collaborative interaction. The dialogue segments appeared more frequently and widely distributed in condition S, while they were shorter in length and rare in occurrence in Condition NS. This indicates that the Shared Stage module enabled the participants to have better co-ordination through frequent interaction. The result about the turns also gives a similar indication. It was found that there was more conversational exchange and turn taking in condition S. Participants had a longer time for verbal interaction in condition S. It can be said that the addition of the Shared Stage module allowed participants to have a more active group environment. In particular, the distribution of dialogue segments and verbal interaction does show the difference in the pattern of the collaborative interaction between two conditions. The context of verbal interaction shows that pairs in Condition NS largely focused on the checking process, while pairs in conditions S focused on the methods and the shared outcome. This indicates that the participants using the Shared Stage window could interact with their partners over more sophisticated issues during collaboration, because the Shared Stage module provided basic awareness of their participant's process. The Shared Stage window was frequently used during the collaborative session showing the usefulness of the window.

With regard to the participants' perception about the conditions, the satisfaction was much higher in condition S. Participants expressed that the user interface of the Shared Stage was easy to understand and the collaborative features improved their satisfaction. Most participants preferred the condition with the Shared Stage module.

6.5 Exploratory results and analysis

6.5.1 Strategies for collaborative modelling

The strategies used for the collaborative 3D CAD tasks have been observed to examine if the Shared Stage changed the way that participants work together. Several strategies of collaborative 3D modelling were identified by reviewing the transcriptions and video recordings. Although the detailed approaches were slightly different from one group to another, the groups' strategies could be categorised into four types: task division, role division, task-role hybrid and tight-coupled collaboration. In this section these strategies are explained and the correlation between the group strategy and the experimental conditions are discussed.

6.5.1.1 Task division

The most common strategy of collaborative 3D modelling was to collaborate by task division. In the task division strategy, participants divided modelling tasks by components and accomplished the work independently. Participants began by analysing the modelling tasks and deciding important configuration settings to be shared, such as scale and locations of integration. Then they divided the tasks according to the components of the model. Individual work was integrated later. Task allocation by components allowed the participants some independence in carrying out the building process. Once the components were divided, decisions about detailed dimensions and build method were made independently, although co-ordination of the components for integration was required in a later phase. For effective group work, tasks must be allocated evenly to all members within the group. The members in a group are then able to finish the task at the same time and accomplish integration. The users' skills with the tools and their preference for 3D modelling techniques need to be considered for task division. This strategy requires well co-ordinated integration in the final phase. Usually there was one person in charge of this integration work, as it was difficult to share this responsibility among group members. An effective management of component integration became important as the number of components increased.

6.5.1.2 Role division

Some groups allocated different roles to each group member. While the task division strategy was focused on the tasks and components of the model, this approach was more concerned with the method of modelling. Participants identified different roles according to the modelling methods. For example, as soon as the task started, one person in a pair became a component creator and the other became an integrator. The integrator directed what should be created and how it should be made, and the 'creator' built the component according to the instruction.

In this strategy, the planner was able to decide upon the entire process and change modelling methods while the modelling task was in progress. Co-ordination was effectively managed in this approach, although sometimes there was a delay because the integrator had to wait for component creation. This approach would be particularly useful when group members have different levels of skill or specialities in modelling. As the model becomes more complicated, the role of the planner becomes critical for an effective management of collaboration.

The inequality of workload could be a problem in this strategy. Integration and planning might need a good insight for the whole modelling method, but might not require a heavy workload. On the other hand, component creation might take a long time if the components are built one by one. For a group consisting of people with similar skills, the issue of assigning roles might not be obvious. There is a possibility that one member may not be satisfied with the group process because the work is not evenly distributed by role assignment.

6.5.1.3 Task-Role hybrid

This strategy lies in between the task division and the role division strategy. The task division strategy was useful for dividing workloads but integration was usually carried out by one person. Consequently, some inefficiency of collaboration occurred as in the later stages tasks were sometimes not allocated evenly to all group members. On the other hand, the role division strategy was efficient only when an appropriate job division was possible. For example, there was little work for the integrator to do in the initial part of the session. The task-role hybrid strategy begins with task-division in the initial stages, and in the later phase they change the strategy into role division. Participants divided tasks in the initial phase and when the final phase was reached, one person became the integrator. The integrator worked on the main structure of the model, checked the progress of related components, and sometimes directed their partner to the change or refinement of the work.

6.5.1.4 Tight coupled collaboration

In this strategy participants collaborate closely from initial phase to integration phase. They start to divide the model into smaller pieces and then work together to build each component as well as arranging them in the current location.

Because the work is tightly coupled, all know the status of the modelling process and can share ideas about the model. It is even possible to discuss very specific problems during the process. For example, in the first task of the experiment, the participants using this strategy started to divide the components into chair, table and drawers. The group decided to build the table together: one person built the tabletop while the other built the legs. The position and the size were decided while the modelling of the table was carried out. The method and the detail properties of the model were changed dynamically. The whole process was more flexible than other strategies.

The ability to allow participants to check their partner's progress during the task is essential for the running of a tightly coupled collaboration strategy. In Condition NS such continuous awareness was difficult to achieve and therefore only groups in condition S were able to take this approach.

6.5.1.5 Correlation between the system and the strategy

Table 6.16 shows the summary of the four strategies mentioned above. Table 6.17 shows the strategies taken by the pairs during the experiment.

Table 6.16 Strategies of pairs

Type of Strategy	Description
A: Task division	Independent component building and integration by one person
B: Role division	To assign different roles (e.g. Component builder and integrator)
C: Hybrid of task and role division	Task-division in the initial stages and role division in the later phase.
D: Tight coupled collaboration	To collaborate closely from initial phase to the integration phase

Table 6.17 Strategies taken by pairs during the experiment

Task1								Task2							
Condition S				Condition NS				Condition S				Condition NS			
P1S	P3S	P5S	P7S	P2NS	P4NS	P6NS	P8NS	P2S	P4S	P6S	P8S	P1NS	P3NS	P5NS	P7NS
A	C	A	D	C	A	A	B	C	D	A	B	A	A	A	B

It can be seen that the strategies varied according to group characteristics, task type and the system used. Firstly, pairs tended to choose the same strategy although the task and condition changed. Five out of eight pairs kept the same strategy in both conditions. It also appeared that task types also influenced the selection of the strategy. In the first task, the model can be easily divided into chair and table components. Therefore, half of the pairs used the task division strategy regardless of conditions. The systems used seem to be another factor in the collaboration strategies. In Condition NS, five out of eight pairs took the Task division strategy, while strategies taken by pairs in condition S were more diverse. It can be inferred that the Shared Stage module provided more flexibility in choosing strategies of group work while the conventional way to collaborate via file exchange confines the strategies to either task or role division. Although there are other factors that influence the selection of the strategies for group work, the system also had a positive influence on the way that the group members collaborated by providing more alternatives in strategy.

6.5.2 Exploratory comparison between the two conditions

In order to examine the differences of overall collaborative 3D modelling processes between the two conditions, three aspects of collaboration, planning, accomplishing individual and shared work and integration, have been examined. For each aspect, the impact of the system to the activities were also investigated.

6.5.2.1 Planning

Planning activities took place at the starting phase of the collaborative session. It involved discussions about the tasks, modelling methods and collaboration. One of the main activities was to understand the task and decide all of the parameters necessary for modelling. For example, participants estimated the scale and the dimensions, in the first task, according to the drawing and then reached an agreement. Some pairs discussed particular parameters of the task at the starting phase, such as the origin and orientation of the model, but others made those decisions only as needed. Most pairs used the grid in the modelling window as a guideline to decide the scale of the model. Some users kept a record of what was agreed, but most of the users did not keep these details. Rather they asked their partner to confirm them when they were not sure. Modelling methods were also discussed at the planning phase of the group session. Other important activities for planning include allocating the tasks or dividing roles. Most groups divided the model of the first experiment task into two parts, the chair and the desk. The model of the second task was divided into the bottom part and the top part.

During the planning activities, the Shared Stage module seemed to affect the group work. Planning activities in condition S were more distributed over the entire session and shorter in the initial phase, while those of Condition NS were more specific and concentrated during the initial phase before individual members started working on their own part of the model. In Condition NS, detail planning was required because they could not have interactive discussion about the on-going task. Such detail planning, however, was not essential in condition S as they could easily consult via the Shared Stage module. For example, the following dialogue segment (P6NS:01-03) shows that the dialogue is quite specific in terms of deciding the dimensions and the scale of the model. Because group members in Condition NS tended to work separately after the planning phase, the mode of collaborative interactions was different from that of individual construction of the parts.

Dialogue Segment P6NS:01-03

M: 7 x 3 the desk.
R: 7 x 3
M: The desk...no the desk is 7 x 3.
R: Oh right.
M: So I'll do the desk yeah?
R: Yeah if you want.
M: The legs are 3.25. It doesn't matter...you do the chair.

R: Obviously not loud enough. What's the height of the desk off the floor?
M: 3
R: Cheers.

M: 2.6...8
R: The height of the desk.
M: Yeah
R: And I go for 3 for the height of the chairs.
M: Yeah
R: Sounds good.

M: 3
R: What for the height of the desk?
M: Yeah
R: Ok so I'll do 3.5 or maybe even 4.

On the other hand, in the dialogue segment of Condition S described below (P1S:01-02) user N and B briefly agree on the work allocation and were able to start modelling without decision on the detailed dimensions and the scale of the component. The relative dimensions were decided as the modelling work progressed.

Dialogue Segment P1S:01-02

N: So what are you going to do?
B: Well ...
N: Ben?
B: Well Nadia I think I will do the .. the chair
N: The chair and I'll do the table.
B: If I finish the chair quickly, I can give you a hand.
N: OK, you can give me a hand making drawers or something. just copy them, OK I will do the table.

B: I think my chair is a bit too small for your table so I'm going to enlarge my chair slightly OK.
N: OK
B: It looks to be about sort of a fifth of the length of the table. That's about a fifth I think. Two squares of your table so ten squares long, so that's about right.

The following example shows how the SSW helped the members to share the work dynamically as the task progressed. In the dialogue segment P4S:04, the work allocation was not strictly fixed before this dialogue. As user K finished his assignment, he asked for additional work. Instantly they both realised that the aerial part was left and user K started building the aerial. Using the Shared Stage Window, participants could start with a flexible plan for modelling. Then they could modify the plan according to the skills of participants, the complexity of model, the system performance and the collaboration strategy. In the dialogue segment P7S:04, user G suggests helping with the drawer by adding the knob components. The division of work was decided as the model was being completed.

Dialogue Segment P4S:04

T: I am doing the whole of the box mainly. Can you see what I am doing?

K: Yes I can. What's left for me to do?

T: The aerial

K: The aerial.. I can do that..

Dialogue Segment P7S:04

G: Ok what are you building now?

R: I'm building one of the drawers. That's about the size of one of the drawers.

G: I'll do the button on the front then shall I?

R: Ok.

6.5.2.2 Individual and shared work

There was a notable difference in the way of co-ordinating individual and shared work. In Condition NS, it was found that participants tended to work separately. Until they had finished the initially allocated work, they focused only on their own work and did not pay attention to their partners work. On the other hand, in condition S, group members frequently move between the individual and shared workspaces. When the individuals finished one part of the model, they wanted to have confirmation from their partner. When they reached an agreement about the size, location and the way to model, they frequently moved between the individual workspace and the shared stage window for further discussion and confirmation. They were able to help partners who were struggling or slow with the work. The interconnection between shared and individual activity took place easily through the Shared Stage Window.

6.5.2.3 Integration

The experimental session showed that it could be difficult to work together for integration using conventional methods. By exchanging files the sender and receiver could not easily provide the related co-ordination support for integration. As a result, most groups in the file exchange condition finished the integration without comprehensive refinement of the final model. On the other hand, pairs in the Shared Stage condition constantly integrated the components while models were being built. Therefore, there was very little work left for integration in the final phase. In addition, they frequently discussed modelling methods throughout the modelling process, all members in a pair had full awareness of the task and the entire process. They knew how the final model was completed, the evolution of the model and the detail modification regardless of the ownership. The Shared Stage module played an important role in improving the integration process.

6.5.3 The role of the Shared Stage module

6.5.3.1 Awareness support

The participants frequently checked the work status of the other group member through the Shared Stage Window. Participants used the Window to see their partner's activities when there was no direct verbal interaction as well as whilst communicating verbally. Because the status of the workspace is always visible, users may automatically know the working status of their partner by examining any changes to the workspace. The hierarchical tree was also used to check the workspace structure of others. In addition, it provides history information of object creation within the workspace, as newly built components are listed at the top of the tree. The following verbal interaction is an example to show how the Shared Stage module was used to support mutual awareness of the workspace status in the experiment session.

Dialogue Segment P1S:07

```
B: I finished my chair. Can you see my chair? Have a swivel round.
Have a look at it.
N: I'm turning it round. Hmm, why is it in the end of the table.
```

```
B: it is..
N: it's in my table.
B: I will group together and put I in position there. So you've got an
idea of where it is so..
```

For example, in the dialogue segment P1S:07, user B finished modelling the chair part and wanted confirmation from his partner as to whether it was appropriate to proceed to the next stage. They both examined the completed parts and refined them through the Shared Stage module. As it can be seen in the dialogue, the location of the chair was not correct, so user N asked user B to relocate the chair. Because both users share the same view from the synchronised camera, the adjustment could be seen by both users. Consequently, the relocation of the chair part could be easily completed.

In the dialogue of P7S:06 07, user G observed their partner moving objects around the workspace and got to know what the partner was building. By being aware of their partner's status, user G could help instantly when user R asked for assistance building the back and the seat of the chair model.

Dialogue Segment P7S:06-07

```
G: You like to drag and drop don't you? I can see you doing it.

G: Are you constructing the chair?
R: Yes I am. I'm doing the chair leg...the four chair legs. So if you
  want to do the sitting part of the chair and the back of the chair
  you may. The back two legs is what you are drawing now. They go
  through the sitting part of the chair and attach onto the back part
    of the chair.
G: Yeah, but they're two separate drawings.
R: Shall I do them as two separate drawings then?
G: Ah it's up to you, do it however you like.
R: Yeah I think I will actually. I'll do it properly.
```

6.5.3.2 Preserving coherence of modelling

Coherence is one of the most important aspects in a collaborative 3D design task. Conventionally, users build components independently and integrate them at a final phase. Final models need coherence in terms of various modelling features, such as dimensions, positions and the level of detail. Sometimes modification of completed parts is required, in order to maintain coherence between parts made by different users. The Shared Stage module allowed participants to preserve the coherence of modelling. Because the Shared Stage module supports ways of examining the relationships between the different parts being modelled, participants were able to consider the coherence throughout the collaborative session. The following dialogue segment (P1S:04) shows how participants maintained the coherence of the model using the Shared Stage module.

Dialogue Segment P1S:04

```
N: Why is your leg in the air?
B: My leg is too high actually. I'll move it down a bit.
B: I just gonna reference some of your bits so that I can see them
with other things.
N: As long as you reference them and don't take them away.
B: I can see where you are now.
```

This dialogue segment took place at quite an early phase of a collaborative modelling session. User N and User B divided the modelling task into two main components, the table and the chair, then they started to build each component separately. After they had built part of the component, they noticed the relative location and the dimension between components were not coherent. They examined the basic relationship from the Shared Stage Module and adjusted the location of the components as the building process proceeded. Consequently, the coherence of the entire work could be maintained from the initial to the final phase.

6.5.3.3 Dynamic exchange of model data

The dynamic data reference and exchange feature of the Shared Stage Window improved the collaboration process. Participants used the referenced objects as a guide to determining the relationships between different parts modelled separately. It reduced the need for verbal interaction to explain the details of what should be considered for the shared work. For example, when one user positioned the location of the table leg, the position of the tabletop was referenced and used as a guide object. Since the colour of the referenced object appears differently in the local workspace, the user could pay attention to building local objects without interference from referenced objects.

On the other hand, object teleporting was effectively used to exchange data between workspaces. For example, in a situation where one user had difficulties with modifying an object to a desired shape, the partner teleported them and tried other methods. They were able to discuss the problems and the existence of alternative methods of solution. Teleporting was being used with more caution because it might interfere with a partner's on-going work. When teleporting was used, participants sometimes wanted to be protected from the involvement of others in individual modelling processes. The dialogue segment P1S:07 presented above also shows that user N expressed that she did not want to be affected by teleporting.

6.5.3.4 Collaborative operation support

The Shared Stage module enabled users to accomplish new types of collaborative operations in the modelling process. One such collaborative operation was where one person guides the modelling process and the other follows their instructions. The Shared Stage Window could be used as a medium for guiding as well as a place for demonstration. Even very specific functions like positioning or transforming objects could be done collaboratively. The following dialogue segment example (PS7:03) shows how participants worked together for such collaborative operations.

Dialogue Segment PS7:03

```
R: X then we want to go Y. The leg's 4 so I want to go 4 high.
G: Right that's in the air now.
R: Don't test it.
G: The base of the table is below the height of the leg.
R: The base of the table is below the height of the leg.
G: IE the leg goes half way...
R: Into the table.
G: Half way through the table.
R: Ok I'll raise it. Yeah I know how to compensate for that actually.
```

G: 0.25 I think.

R: Yes that's right. So I'll take that and I want to move it...then 0.25 high yeah?

G: yeah

In this dialogue segment, user R and G were both making the table together. The detailed dimensions and the location of the table legs were decided collaboratively through the Shared Stage module. User R controlled the objects while user G provided the direction for the activity. On the other hand, the following example(P8NS:04) shows the difficulties of such guiding and control activities in Condition NS.

Dialogue Segment P8NS:04

H: Marc?

M: Yeah sorry.

H: Are these units actually...these little squares on the screen, do they actually count for one? 'Cos I've said...

M: Yeah, each square is one...

H: Yeah, I've said it to do a thing 12 wide, which means 6 in each half surely?

M: Yeah

H: And there's only 5 and a bit in each one, there's not quite 6.

M: Ah!

H: So I'm not quite sure what's going on there.

M: That's very strange.

In this dialogue segment, user H was having difficulties understanding why the width of the box object (table top) was not as expected. He created a box, and moved it to the wrong place by mistake. Although it was a simple mistake and could be easily understood by others, he spent quite a long time attempting to understand this problem. He was trying to explain the problem to his partner, but the partner could not easily understand the situation from only a verbal description. Therefore, the partner could not provide any help although he knew the problem existed. With the Shared Stage Window, the problem could have been solved more easily by working together in the shared workspace.

6.5.4 User interface issues of the Shared Stage Window

6.5.4.1 The limitation of interacting with a 3D workspace on a 2D screen.

One of the major problems in conventional 3D applications is that they are producing artefacts in the 3D form, but their tools are only capable of managing 2D. This problem is increased in a collaborative modelling situation, where users not only have to interact with their own 3D objects but also those of others in the Shared Stage module. The 3D representation needs to be easily understood by all members within a group. The interaction with 3D objects in the shared workspace needs to be simple to minimise the limitation. In the Shared Stage module, however, the navigation of the 3D workspace required particular mouse and keyboard operations as with conventional 3D applications. Awareness of a partners 3D work area is also important to collaboration in shared 3D workspaces. Further consideration should be given to ways of providing natural 3D interaction and 3D workspace awareness information, such as the location of object or user views in a 3D environment.

6.5.4.2 The problem of access control in the local workspace

As can be seen in the dialogue segment P1S:07, users found that teleporting could interrupt their individual modelling work. The Shared Stage module allowed users to move the objects between woıkspaces using teleporting functions. However it also caused confusion to the participants when new parts were suddenly created in the local workspace, or parts of the local model were suddenly transferred to another workspace. Therefore, sometimes users wanted a secure protection of local models from the involvement of other users.

This problem can be solved by providing more awareness or notification support in the shared workspace or developing new access control mechanisms. The first way to address this is to support an awareness mechanism. If a person who runs a teleporting function knows how his action influences the partners work, he will wait or ask the partner about the appropriateness of the operation. A notification mechanism could also provide information as to whether certain functions were executed locally or remotely. Another way of addressing the problem is to add a direct access control mechanism for all objects. For example, a mechanism similar to UNIX file permissions might be introduced. An objects' permission status is defined as accessible or not-accessible to group members. When users want to modify objects, they should get permission before performing any action on the objects.

As there was instant sharing of 3D models created in the private Syco3D window with all participants in a session, confidentiality and privacy issues could be problematic. The Syco3D system is, however, intended for a small group collaboration, in which privacy and confidentiality may not be a critical problem during a real-time session. However, if the system is used by a large team involving professionals from a different background or organisations, ways of addressing the confidentiality problem need to be addressed.

6.5.4.3 The lack of workspace recovery and recording support

Collaborative processes involve extensive modification and refinement. A series of sessions may be required for the completion of a complex modelling project. To support this evolutionary process, an effective mechanism of recording data for group use is required. Because the Shared Stage module does not incorporate the function of recording intermediate results during the modelling process, participants could not go back to the previous status of the workspace. In the current version of the system it was necessary to recording the status individually. Information specific to a multi-user environment, such as ownership, access level to models, user state change during a session, needs to be incorporated in the data recording. Efficient ways of holding data for intermediate progress also need to be invcstigated since users need to review the current or past status of the workspace.

Sharing information about workspace evolution is also important. Part of the workspace history is displayed in the hierarchical tree by showing objects on the top of the tree as recently created or modified parts. This feature of the Shared Stage module allowed members in a group to gain an overall picture of the workspace history and its process. However, further support of information about its evolution needs to be considered.

6.5.4.4 Difficulties in finding objects in the shared workspace

As object selection is fundamental to the manipulation of 3D objects in the workspace, finding and selecting objects of interest is one of the most frequent tasks in the shared workspace. Easy search and selection is also connected to efficient referencing and teleporting of 3D objects. In the experimental system, finding the correct objects was accomplished in the hierarchical tree window. By pressing the nodes in the tree, the object is highlighted with a red outline. In several sessions, users expressed difficulties in finding the object required. They sometimes had to scroll through the hierarchical tree window because there was not enough space to display all the information about the parts as the model became more complex. The following dialogue segment example (P4S:18) shows the difficulties of object finding and searching.

Dialogue Segment P4S:18

```
T: OK. well I am gonna have a look at what you are doing. It seems
roughly in a right place. Which number is you base? 22?
K: I couldn't tell to be honest.
```

One of the problems in finding the correct object occurs because the nodes are indicated as numbers in the Shared Stage Window. It is difficult to find a direct connection between nodes and the actual object in the modelling view. Names can be implemented to indicate objects for the nodes and to facilitate searching processes, but users still have to specify the name every time an object is created. Therefore, a simple mechanism for searching and selecting objects in the shared 3D modelling space is essential.

6.5.4.5 Representation of models in the Shared Stage

Representation of models in the Shared Stage should reduce the visual complexity in the Shared Stage, while supporting efficient shared understanding among different designers. 3D models are shown as a wireframe representation in the owner's colour. The effective use of colour needs to be further considered. Colours showing ownership information should be carefully selected as the human perception of colours is limited and may depend on the context of their use. This means that the number of useful colours for ownership labelling is limited. Problems may also occur when models need to be represented differently, for example, according to their functions or related assembles. Another issue related to representation of models in the Shared Stage is concerned with ways of supporting a shared understanding of models among collaborating designers while maintaining customised representation for individual designers. For example, different levels of representation details may be applied to enable a designer to focus on the parts associated with his/her models in the Shared Stage module.

6.5.4.6 The lack of the screen resource.

Efficient management of screen space needs further consideration. During the experiment, the addition of a Shared Stage module limited the way users examined different views of the workspace. When users were engaged in a private Syco3D window, the Shared Stage module blocked part of the local windows. One way of addressing the screen space is to automatically resize the Shared Stage Window dependent upon user focus. For example, the Shared Stage Window may be kept small for providing essential workspace awareness information, but automatically resized into a larger window when extra attention is given to the Shared Stage. Alternatively, the Shared Stage module can be incorporated into the perspective view of the Private Syco3D window with a mechanism to select the mode of sharing.

6.5.5 Summary of exploratory analysis

Several strategies for collaborative 3D design, including task division, role division, hybrid and tight-coupled approach, were identified. As the Shared Stage Window provided a more flexible collaborative environment, participants could choose a variety of strategies to accomplish group work, while conventional systems confine the strategies to either task or role division.

Two experimental conditions were compared by observing three aspects of collaboration: planning, accomplishing individual and shared work and integration. The shared stage module changed the way participants planned the task. They could start to work without specific plans using the Shared Stage module. It was possible for them to modify or improvise the plan during the task. Planning activities in Condition S were more distributed over the entire session and shorter in the initial phase, while those of Condition NS were more specific and concentrated on the initial phases. With regard to the co-ordination of individual and shared work, the Shared Stage Window allowed participants to have a more dynamic and frequent transition between individual and shared work. Many inefficiency problems found in the Condition NS, such as repetitive integration and uneven job allocation during integration, were addressed in condition S. Participants were able to solve the problems together. Some new collaboration approaches for 3D design tasks were found in condition S.

The Shared Stage module was used as an important resource for collaboration. It was used as a main platform for awareness support and co-ordination of ongoing models. Allowing flexible planning and supporting dynamic collaboration, it also served as a data exchange medium and a place that keeps the final completed model. Various collaboration approaches could be adopted using the Shared Stage Window. Some limitations were also found while using the Shared Stage Window. Since Shared Stage Window is another 2D projection of a 3D environment, 3D interactivity is confined, as with conventional 3D design tools. Teleporting functions sometimes made participants feel their individual workspace was insecure. The lack of workspace recovery and record support also needs further consideration. Difficulties in finding and searching for objects in the shared workspace were also experienced. The addition of the Shared Stage module limited the way that participants used the screen resource. Nevertheless, the exploratory analysis indicates that by providing an interactive and easily accessible shared 3D workspace during 3D design tasks, the Shared Stage module made a positive impact on the collaborative 3D design activities.

6.6 Conclusion of the usability evaluation

The experiment demonstrated the usability and usefulness of the Shared Stage for real-time collaborative 3D design activities. However, in order to generalise the result, further refinement may be needed for the measures used in the analysis based on the usability evaluation standard (ISO, 1998). Consideration should also be given to the specific issues relating to the evaluation of groupware systems as those suggested by Grudin (Grudin, 1994a). Based on the result of quantitative and explorative analysis, the following conclusions can be made.

- The Syco3D system was successfully used as an evaluation system incorporating the Shared Stage Module. Moderately complex 3D modelling tasks could be accomplished.
- The interactions between participants during collaborative 3D design activities were changed by the Shared Stage, as it initiated more active interaction between participants.
- Satisfaction about the outcome and collaboration process could be greatly improved by introduction of the Shared Stage. However, a direct contribution of the Shared Stage to the quality of outcome was not detected.
- It was found that the use and role of the Shared Stage varied in different phases and tasks. Smooth co-operation of the Shared Stage is required for tasks with which individual workspace should be maintained
- Participants could use the Shared Stage for many different roles in the phase of planning and integration. The usefulness of the Shared Stage is demonstrated by the fact that the Shared Stage Window was used frequently throughout the collaborative session.
- The collaborative features provided through the Shared Stage could be comfortably used by participants for collaborative 3D CAD activities.

6.7 Chapter summary

This chapter presented a usability evaluation of the prototype system. The main objective of the usability evaluation was to examine if the Shared Stage framework and the research prototype were feasible when used by real users. It also aimed to evaluate functional capabilities and examine the impact of the system on the collaborative 3D CAD process.

A laboratory-based experiment was designed and qualitative and exploratory analysis was carried out. For comparative evaluation, a pair of designers carried out experimental 3D modelling tasks using two different versions of the system; the single-user version and the original collaborative version of the Syco3D system. The outcome from the group task, collaborative process, and user satisfaction were analysed using quantitative methods. Collaborative strategies, major collaboration activities and roles of the Shared Stage module and breakdown caused by the system were analysed using exploratory methods.

The results suggest that the shared module is a valuable addition to the group task, and allows users more flexibility in the strategy of group work and dynamic interaction with individual and shared 3D workspaces. Although results could not show direct influence of the addition of the Shared Stage, the user response from questionnaires, interviews and observational analysis indicates that it was valuable in supporting the dynamics and flexibility of group activities. The exploratory analysis showed that the Shared Stage module provided an interactive and easily accessible shared 3D workspace during 3D CAD.

Chapter 7

Discussion and Further Issues

7.1 Introduction

In the previous chapters, the framework for a real-time collaborative 3D CAD system was demonstrated and evaluated through a usability experiment. This chapter discusses the results of this investigation in relation to previous studies. Further issues are also discussed in order to extend the real-time collaborative 3D CAD system to a higher level of a computer based collaborative design environment. Four related issues are considered in particular: the relationship between a real-time collaborative 3D CAD system and collaborative design environments, 3D interaction techniques in a shared 3D workspace, using the World Wide Web as a platform for the real-time collaborative 3D system and further support of 3D workspace awareness.

7.2 Relation to previous studies

The results from three parts of this investigation, the preliminary study (Chapter 3), the conceptual framework and the prototype system (Chapter 4 and Chapter 5) and usability evaluation (Chapter 6), are discussed in relation to the previous studies reviewed in Chapter 2.

Firstly, the preliminary study presented in Chapter 3 supports previous design process models (Jones, 1970; Alexander, 1964). It also demonstrates the feasibility of current collaborative technologies for a distributed team design project, as examined in other studies (e.g. Maher et al., 1997). The preliminary study used a similar setting to those of earlier investigations of team design activities (Cross et al., 1996; Tang, 1991) and computer mediated collaborative design (Scrivener et al., 1993; Maher et al., 1997). These studies used detailed protocol analysis in order to analyse design activities. Transcribed conversations were segmented and coded for quantitative analysis. Although earlier work suggested some implications for new collaborative technologies based on the protocol analysis, the conclusions were too abstract for application to the investigation of novel tools. While they focused on microscopic analysis of interactions between designers, the preliminary study attempted to identify novel collaborative technologies to be incorporated in a co-located or distributed collaborative design environment. The findings of this part also support Tangs (1991) study emphasising the role of non-verbal communication during collaborative design session. It also highlights the smooth interchange of design media. In order to improve the preliminary study within the scope of this thesis, the study may need to focus more on how designers communicate each other about 3D aspects of design idea. The use of 3D artefacts, mockup, or CAD models can be investigated in a collaborative design session. The investigation may be carried out as a field study involving teams of designers in actual design workspaces, for example design consultants or product design departments of a company.

Many issues of shared 3D workspaces discussed in Chapter 4.4 have been also identified by other studies. Gutwin (1997) emphasised the importance of workspace awareness, which is concerned with the information of what other participants are doing in a collaborative session. Tang (1991) suggested that sharing of intermediate processes is useful for collaborating designers in team design process and highlights the issue of incorporating individual and shared workspaces. Commercial CAD applications stress the consistent user interface and the dynamic data exchange (Hewlett Packard Company, 2000).

The Shared Stage framework of this thesis can be compared to other real-time collaborative 3D environments. Considering that the fundamental concept is to incorporate both individual workspaces and shared workspaces, the framework holds a common concept of 'private and public workspaces' (Stefik et al., 1987b). The Shared Stage framework extends the general concept of supporting 'public and private workspaces' into 3D workspace by highlighting some unique issues for designers. The two main components of the Shared Stage module in the prototype system, Synchronous Stage View and Data Structure Diagram, address these 3D issues. The Data Structure Diagram of the Shared Stage takes a similar approach to the Scene-Graph-As-Bus approach, which uses a common scene graphic as a means of exchanging data between heterogeneous stand-alone 3D graphical applications (Zeleznik et al., 2000). The difference is that the Scene-Graphic-as-bus approach is an underlying structure which focuses on the data exchange, and does not provide user interface mechanisms for collaborative interactions. On the other hand, the Shared Stage uses a Data Structure Diagram as an effective means of collaborative interaction during real-time collaborative 3D CAD.

Previous studies have attempted to address the issues of 3D aspects in shared workspace using traditional desktop environments. In particular, Teledesign (Shu and Flowers, 1992), Cspray (Pang and Wittenbrink, 1997) and Co-CAD (Gisi and Sacchi, 1994) allow designers to accomplish similar features of real-time collaborative 3D construction and modification, as the Shared Stage framework provides. These have largely been built on the desktop based 3D tools. The desktop metaphor, however, may not be efficient for incorporating collaborative features in 3D workspace as it assumes that activities take place in individual desktops having minimal awareness information of others. Other research employs a place metaphor, in which users are represented as avatars and move around collaborative virtual environments (ActiveWorlds, 2000; Stenius, 1996). Although collaborative virtual environments may be efficient for social interactions, engaging in sophisticated tasks, such as collaborative construction or modification of 3D models, can be problematic due to the complexity of interfaces. The Shared Stage combines both desktop and place metaphor in order to address issues arising form the previous approaches. For individual workspace, it allows users to keep traditional CAD desktop environment, while for collaborative 3D virtual workspaces, it provides a shared place, in which collaborating designers interact each other having special collaboration features.

In terms of the usability evaluation, the investigation was carried out as a typical laboratory based experiment. Most of the previous studies of real-time collaborative 3D CAD tools only provided an initial exploration with new tools (Shu and Flowers, 1992; Pang and Wittenbrink, 1997; Gisi and Sacchi, 1994). The primary difference of this work from previous studies is that the proposed framework is examined through a more detailed user evaluation, incorporating quantitative and qualitative analysis of collaborative process using the novel design tools. The findings of the user study supports Gutwin's (1997) study, which showed the importance of workspace awareness. The usability evaluation of this investigation extends the issue of the awareness support into shared 3D workspaces. The difficulties of evaluating groupware systems in a controlled environment (Grudin, 1994a), have been also perceived in the usability evaluation.

There are also some aspects to be considered in order to enhance the validity of usability evaluation. It can be suggested that the complexity of the experimentation tasks may be adjusted. The tasks used in the evaluation were chosen by considering the performance of the prototype system, the participants' acceptance of the new tool and the complexity of measures. One can argue that the tasks are too simple to reflect actual collaborative 3D CAD activities. Secondly, the usability evaluation only investigated a situation involving collaboration between two designers. The collaboration between more than two designers may raise different technical and user interface issues of the tool. Next, analysis of the process largely focused on the observation of computer screens and the conversation of designers. The methods of objectively analysing the collected data also need to be considered. For example, it may be necessary to analyse the data without knowing which condition the data come from, in order to avoid an initial assumption of the impact from the new tool. Ways of capturing all interactions during a collaborative session may provide more accurate interpretation of the designers' activities using the new tools. These aspects need to be taken into account for examining further impacts of the Shared Stage framework.

7.3 The relationship between the real-time collaborative 3D CAD system and collaborative design environments

In Chapter 3, four categories of collaborative technologies were identified that should be integral to the collaborative design environment: technologies for resource sharing, communication, process management and co-working. For a successful collaborative design environment, the real-time collaborative 3D system should be efficiently integrated with other collaborative technologies.

In particular, in order to extend the system to a distributed situation, consideration should be given to ways of incorporating real-time collaborative 3D design tools with other inter-personal communication tools, such as video and audio conferencing systems. One way to include interpersonal communication features is to incorporate video and audio communication facilities within the real-time collaborative 3D CAD system. For example, video and audio communication channels can be a part of the Shared Stage module. Figure 7.1 shows a mockup display of such features in the Shared Stage module, which can be implemented in the current Syco3D system. Video images

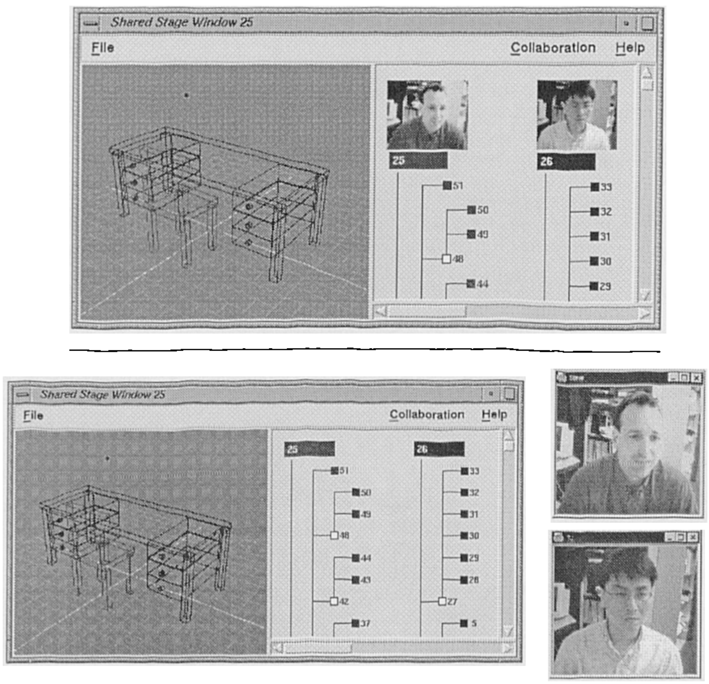


Figure 7.1 Mock-up display for the integration of video conferencing features in the Shared Stage module. Internal (top) and external (bottom)

can be linked with the user nodes of the DSD view (Figure 7.1, top). This approach requires each co-working tool to be used in a collaborative design environment to have basic communication facilities as a fundamental feature. Another possible approach is to provide permanent video and audio communication facilities alongside co-working tools (Figure 7.1, bottom). This approach assumes that communication facilities should be a separate channel in a collaborative design environment. The integration of different kinds of collaboration tools with a real-time collaborative 3D CAD tool should be further investigated.

In the meantime, another issue to be considered with regard to collaborative design environments is the opportunity of extending the system to multi-disciplinary collaboration projects. In this research, the focus has been on team activities involving designers having similar skills and backgrounds. This means that the Syco3D system would be most useful within a group of designers who are equipped with homogeneous CAD environments. However, it is important to consider the same degree of collaboration support for multi-disciplinary design teams. It is widely acknowledged that design teams should include participants from different disciplines, organisations and cultures because the creation of innovative designs may require specialists from a variety of disciplines. Tools to support such integrated activities should reflect the nature of multi-disciplinary design teams (Roseman and Gero, 1996; Haymaker et al., 2000).

One of the issues to be addressed in a multi-disciplinary collaborative design environment is achieving a shared understanding between specialists from different domains (Rosenman and Gero, 1996; Rosenman and Gero, 1997; Saad and Maher, 1996). As illustrated in the Synchronised Stage View of the Shared Stage module, shared representations of 3D data can be used in a situation where a number of team members have to view and discuss the same 3D CAD model simultaneously in the same or different locations. In order to extend our system to multi-disciplinary design teams, it is important to note that sharing the 3D visual representation alone may not be sufficient to a multi-disciplinary team project, because specialists from different domains may interpret the same 3D data differently. A mechanism should be investigated to intelligently translate the changes of one domain to those which are meaningful to other domains. For example, the translation of representation can be useful in a situation where industrial designers and electronic engineers work together to create digital equipment. When a change to the overall shape is proposed by an industrial designer, the impact sometimes affects the internal components of the equipment, such as PCB size and the arrangement of electrical components. The electronic engineer is mainly concerned with the impact the change of the shape has on the internal arrangements. A mechanism to efficiently transform representational information of 3D data to appropriate semantic information would improve the user interface of the real-time collaborative 3D design tool for multi-disciplinary collaborative design environments.

In order to support multi-disciplinary team projects with real-time collaborative design tools, the extra benefits should be acknowledged by all the users in the team. It is also important to consider some features that empower collaborative design tools to go beyond the face-to-face situation. Collaborative design tools may become successful when they can provide extra benefits to group members who are all co-located (Hollan and Stornetta, 1992). Sometimes, anonymity may be useful to provide objective opinions about group decisions (Bannon, 1997). Automatic archival could be another useful feature empowering collaborative design tools even in a co-located situation (Roseman and Greenberg, 1996a).

7.3 3D interaction techniques in shared 3D workspaces

The usability of real-time collaborative 3D design tools can be improved by considering more natural 3D interaction techniques for collaborative situations. A variety of interaction techniques have been used with existing single user 3D CAD systems. In particular, 3D interaction devices, such as 3D digitizers and 3D laser input systems are becoming common. Stereo or head mounted displays are examined as an another type of output system. Such interaction devices can provide a smooth interface between the real world and the 3D environment when tasks involve 3D artefacts and environments. However, because of the bulkiness and expense of such systems, most CAD systems use 2D based input and output devices, such as a mouse, digitizer and flat display monitor. Because designers tasks typically involve 3D artefacts, the limitations of 2D based interaction methods of computer based tools, have been acknowledged as one of the main problems of single user 3D CAD systems. Similarly, in the shared 3D workspace, these input and output devices limit interactivity between designers or between designers and computer based collaborative design tools. Once a desirable interaction technique is achieved, the user interface of real-time collaborative 3D design tools may be greatly improved.

One technique used to improve the user interface is to employ Virtual Reality(VR) environments. In a VR environment, computer generated 3D visualisation is used to represent the 3D environment. Based on the level of immersion that VR provides, it is possible to classify VR systems into three difference types: Immersion Virtual Reality(IVR), Fish Tank Virtual Reality(FTVR) and Mixed Reality(MR). As reviewed in Chapter 2, in the IVR environment users feel that they are fully immersed in the virtual world. This is usually achieved through a head-mounted display which displays images to one or both eyes through the use of small screens located on or near the head. FTVR is a less immersive form of VR in which the real world can be intermixed with the virtual world. This can be achieved with a regular CRT monitor and head-tracking or stereo display. Mixed Reality also overlays virtual 3D images onto real world images but without using the computer monitor. This can be achieved through head-mounted displays or glass-type see-through displays which are transparent but can also overlay computer generated 3D images on the screen. The users can see the real image through the transparent display which overlays computer generated 3D virtual images on the transparent display. Sometimes images captured from video cameras located near each eye are used to overlay computer generated images because overlaid images on the transparent display may not be opaque. Overlaying digital video images with computer generated virtual images may be easier because both are easy to process digitally. The incorporation of inter-personal space and 3D shared workspace may also be achieved. However, differences of viewing angle and visual perception of video camera images could be a problem. For example, it is necessary to adjust the location of the virtual image, in order to achieve the correct stereoscopic views.

Collaborative Virtual Environment(CVE) is another 3D interaction method, employing immersion VR techniques. However, since the Collaborative Virtual Environment is more appropriate to social interaction in cyberspace, the feasibility of using CVE for collaborative 3D design activities remains in question. Bulky and expensive devices have created further obstacles to the practicality of VR systems. A new 3D interface technique should address the problem of 3D spatiality (Norman, 1998).

The Mixed Reality technique can be used in the Shared Stage module; Figure 7.2 shows an approach that employs this technique for collaborative 3D interaction. This setting uses glass-type see-through displays *which can overlay computer generated 3D images* with real world images. The location of the eye can be calculated from locator sensors or from the relationship between predefined reference points in the real world and output coordinates. Then the correct angle of the virtual 3D image can be generated and overlaid on the display. In the co-located situation, where the collaborators wear see-through displays during the collaborative 3D CAD session they will see each other without difficulty. Gazing and hand gestures can be easily supported during the session. This method may be quite natural to designers since it resembles the environment of a traditional design studio, when a group of designers or engineers discuss or work around 3D clay models. Moreover, this setting can be extended to distributed collaborative 3D design environments by adding the representation of the participants' eyes and locators in the overlay images of each site. Since all the 3D location data of the display, the locator, and virtual objects are maintained, it is possible to transmit this data to other sites. In the remote site, the representation of participant's eye and input locator can be re-calculated according to the participants' eye location. Designers in remote sites are able to know where other participants are located, what they are looking at and what they are doing with the locator. Many technical issues, such as the frame rate of display, accurate overlay techniques and 3D locator interfaces should also be addressed and ways of incorporating these features within the Syco3D system require further investigation.

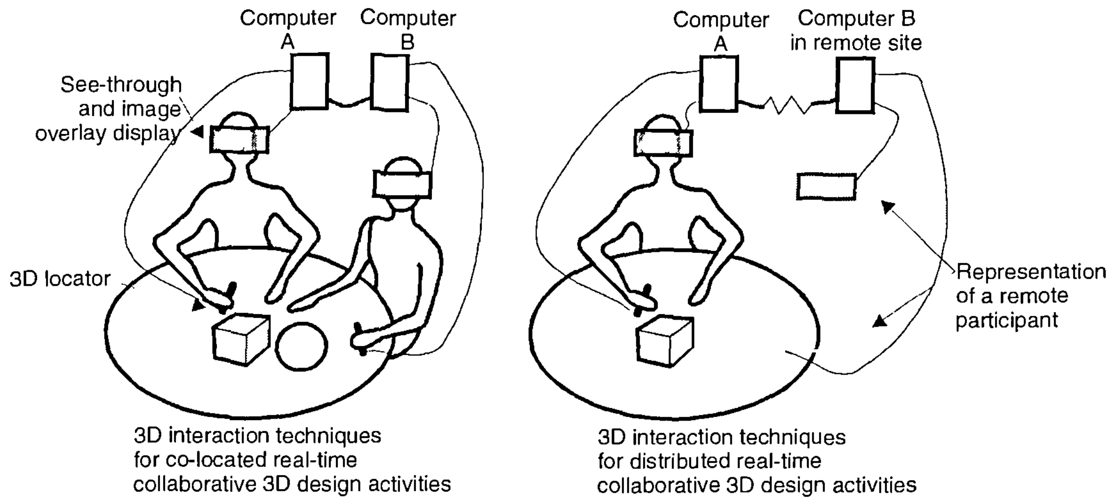


Figure 7.2 Conceptual illustration of a novel 3D interaction devices for real-time collaborative 3D design tasks.

7.4 Web-based real-time collaborative 3D systems

The World Wide Web(WWW) has successfully merged the concepts of hypertext and networked information to provide an easy but powerful global information system, based on two public and simple standards: the HyperText Transfer Protocol(HTTP) and the HyperText Markup Language(HTML). The architecture of the WWW has become established as an important platform for developing, deploying and evaluating CSCW applications. Considering a number of advantages that the World Wide Web currently provides, it is important to consider ways to use it as the base platform for a real-time collaborative 3D design tool.

There are many reasons for the success of the Web, and its attractiveness for users and developers of collaborative applications. Dix (1997) highlighted the following: a core initial user community, the integration of existing information, the use of de facto standards, the spanning of organisational boundaries, and a software platform which is public domain, cross-platform and extensible. In addition, it is considered that the web has reached a point where the benefits outweigh the costs for the user. These advantages have made developers of collaborative applications interested in the Web as an important platform for CSCW development.

In addition, the Web incorporates VRML as an infrastructure and conventions of 3D cyberspace. VRML 1.0 is a subset of the Inventor File Format (ASCII) with some additions to allow linking out to the Web and inclusion of other URLs. In order to describe a 3D scene, VRML provides ways to define geometry, transformations, attributes, lighting, shading and textures. VRML scenes may be viewed within VRML enabled browsers. The JavaEAI (External Authoring Interface) provides functionality for Java applets to interact with VRML worlds. It is possible to perform high-level modelling by combining the VRML and JavaEAI standard.

Although VRML shows potential for defining 3D cyberspace in the WWW, it is expected to take some time for VRML to be accepted as a popular medium for shared 3D virtual space. The first reason is that, for many, building a quality 3D cyberspace with VRML is difficult. The success of the Web has been based on the fact that it provided easy authoring methods. Users are able to build a simple Web page using a text editor and basic HTML code. However, building VRML scenes requires the use of a complicated 3D modelling application which requires a special 3D skill. Also the user interface of VRML is difficult and unnatural to navigate in 3D cyberspace. Furthermore, since there is no generalised user-interface standard for interacting with VRML sites between different VRML browsers users must learn different user interfaces. Finally, VRML has limitations to becoming a data standard that can be used in the manufacturing and design domain. The specification of VRML is not versatile enough to represent all the features of other 3D data standards such as STEP or IGES.

In addition to the limitations of VRML, there are also limitations in the WWW architecture, which make it difficult for VRML based Web systems to provide a platform for real-time collaborative 3D systems. The basic client-server architecture of the WWW is sufficient for serving static pages of information or a 3D VRML scene from fixed locations in the remote file system. However, in a situation where the available information changes frequently and interactively, the basic server needs to be extended. Since the protocol it uses is inherently stateless, the Web has not been accepted as a suitable platform for synchronous collaborative applications (Trever et al., 1997). The information is stored between requests and therefore the server (or application) does not have data on the page that the client is currently browsing. The client invariably sees just a snapshot of the application's state. If something changes in the application, there is no way for the user to know this until he reloads the page. Therefore, co-operative tools and applications which require any significant degree of synchronous interaction between users or applications cannot be supported by the current Web architecture.

To provide special synchronous support within the current web architecture, some browsers introduced 'cookies' to maintain the state and 'javascript' to move some computation from the server to the browser. Another solution is to provide additional infrastructure supplementing Web-based applications, for example by using the Common Gateway Interface. Ways of addressing essential real-time groupware design issues, such as floor control, session management and awareness mechanisms, should be investigated within the Web structure. The collaborative 3D CAD environment involves representing, building, and editing 3D virtual objects within the application. More robust 3D data standards for the web should also be implemented in order to manage dynamic interaction of real-time 3D CAD collaboration.

7.5 3D workspace awareness issues

Workspace awareness plays an important role in the usability of shared application tools. In the Syco3D system, some workspace awareness features were provided in the Synchronised Stage View and Data Structure Diagram. However, further exploration of awareness support unique to the shared 3D workspace needs to be accomplished.

Awareness is related to implicit communication, such as indirect gestures and information about other people's environment. This information helps people to establish common ground, co-ordinate their activities, and avoid surprises. Since awareness can be at odds with concerns about privacy, it is important to find ways of achieving awareness support while addressing these concerns.

There are three broad means to provide the workspace awareness support: embodiment, actions, and artefacts (Gutwin, 1997). Workspace embodiments can convey information about who is in the workspace, where they are and what they are doing. Telepointers, viewports, avatars and video embodiments may be created to support such awareness information. Second, it is possible to make actions more perceivable in a shared workspace. Example techniques are to provide local visual feedback to others during direct-manipulation actions (such as the temporary 'rubber-band' rectangles used to select multiple objects), to display intermediate steps in carrying out symbolic actions (such as menu choices), and to stretch out an instantaneous symbolic command and make it more perceivable. Third, artefacts can also be used as a means of workspace awareness support as these display information about whether they are in use, who is working on them, and what has happened to them in the past. Colours, texts or portraits connected with the artefacts can provide such information as authorship, permission and editing status. The usefulness of some of these techniques has been examined in the 2D shared workspace.

In the Shared Stage module, techniques used to show who is in the workspace, where they are and what they are doing, such as telepointers and 2D viewports, are inefficient representations of the 3D environment, because this provide limited information about the 3D space. One way of supporting such 3D information is to create 3D embodiments. Since 2D telepointers cannot provide accurate locations in the 3D workspace, the implicit representation of such data may be difficult in typical perspective or orthographic views. The embodiment of camera views and locators as icon objects in the 3D environment may be useful in this situation. In order to represent one's view angle, direction and target point, simple representation of these characteristics are required because these embodiments should be easily identifiable in complex 3D environments. In the Shared Stage Module, they can be incorporated to provide better workspace awareness information. Figure 7.3 and Figure 7.4 shows a mockup display that might be expected if these embodiments are used in the Shared Stage module. These embodiments should be carefully designed to minimise the complexities and to reduce perceptual loads of interpreting the meaning of the embodiments.

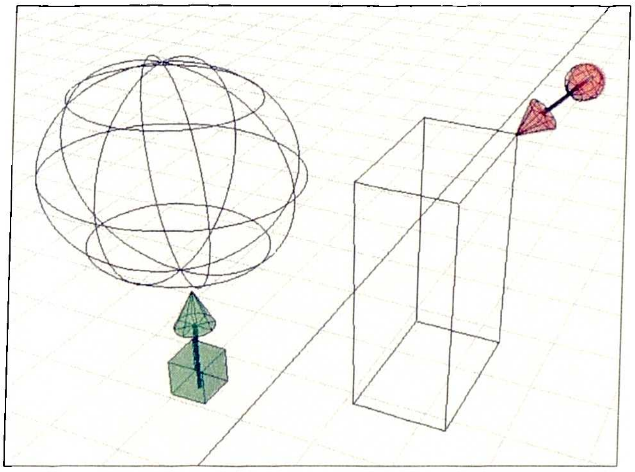


Figure 7.3 Mock-up display of a 3D locator embodiment

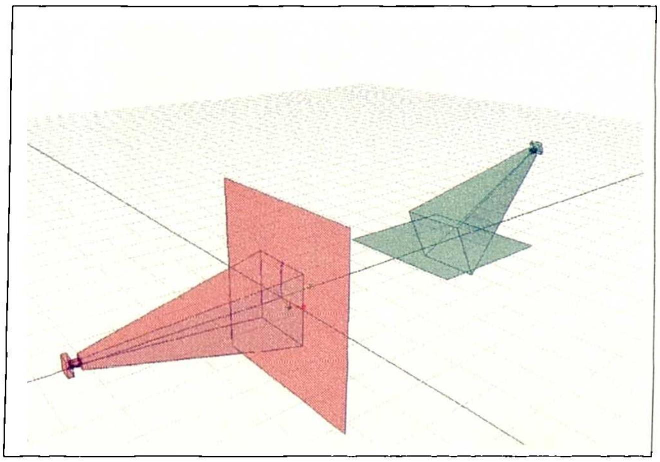
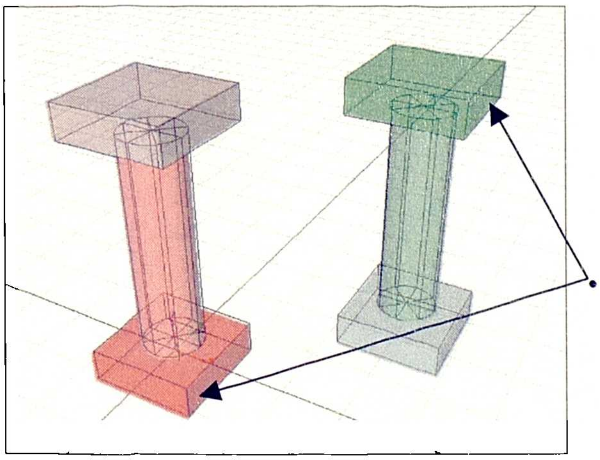


Figure 7.4 Mock-up display of embodiments for user views and work areas

Workspace artefacts can be used as a medium of 3D workspace awareness support For example an object age feature may be implemented in the Shared Stage module. Object age is information showing when the object was created and modified. If such information is implicitly represented in the shared 3D workspace, users may easily see where other participants have created objects, and what kinds of objects have been recently constructed and changed. The tasks accomplished by group members can also be implied through object age information. It may be represented by changing the colour of objects over time. When objects are created, the colour of the object may be displayed in its owner colour. The object colour changes to a darker or less pure colour, within the same colour range, as time goes by. Eventually the object colour becomes a fixed colour. For example, when designer A creates an object, the colour of the object is pure red, which could be object's owner colour. As time goes, the pure red may become greyish. When other people look at the workspace, pure red can be easily distinguished as a recently constructed object belonging to designer A (Figure 7.5). This will provide useful information about what designer A has been doing.



- Objects are originally represented by user colours Young objects are shown in a pure colour while old objects in a faded colour.

Figure 7.5 Awareness support using the object age feature

Because most editing operations include selection processes, awareness support for object selection is also important to the implicit understanding of what object construction and modification operations are taking place. By implicitly showing what objects are selected and by whom in the shared 3D workspace, each participant may anticipate what will be happening in their workspace. Selected objects can be represented in particular colours or the Data Structure Diagram can be used to show activated object groups. This can be connected to the log of commands being executed in one's own workspace. For construction operations, the log of commands and important 3D co-ordinates can be used as important awareness information as to what is being produced in the workspace.

7.6 Chapter summary

In this chapter, the results of this investigation were discussed in relation to previous studies. The preliminary study attempted to clarify the issues of collaborative design environment rather than carrying out a detailed analysis of team design activities. The shared stage framework combined the desktop and place metaphor in order to support the sophisticated 3D design activities in a collaborative environment. Usability evaluation took both quantitative and qualitative methods to investigate the impact of the novel tools in a dynamic multi-user situation.

Further issues relating to ways of extending the real-time collaborative 3D design tool were also discussed. These issues included the relation of the real-time collaborative 3D design tool to collaborative design environments, 3D interaction techniques in a shared 3D workspace, the use of the World Wide Web and VRML as a platform for the real-time collaborative 3D system and 3D workspace awareness issues.

In order for the real-time collaborative 3D CAD system to be fully used in a distributed collaborative design environment, the system should be smoothly integrated with inter-personal communication tools. Asynchronous collaboration and automatic archival might be useful features within a multi-disciplinary team environment. It was pointed out that in these settings it is important to share the representation of the 3D view and the 3D semantics as well as translating the semantics to meaningful data for a specific discipline. Since specialists from different domains may make different interpretations of the same 3D representation, real-time collaborative tools should take into account shared understanding and filtering of the information when used by a multi-disciplinary design team.

The issues of natural 3D interaction techniques were discussed because the tools should provide a natural user interface for the tasks required in the shared 3D workspace. Mixed Reality techniques were introduced as a way of addressing 3D interaction issues for real-time collaborative 3D tools in a co-located or distributed situation. By using these MR techniques, it may be possible to build a collaborative 3D environment similar to a traditional design studio setting for group work, with a group of designers working around objects with a full 3D interface.

Web and VRML issues were also discussed showing the strengths and drawbacks when applied to real-time 3D CAD systems. Despite the many advantages of the web, the stateless nature of the current web architecture may be problematic when it is used as the main platform for real-time groupware systems. 3D interactivity is also difficult to achieve since the interface should be built into web browsers. Current VRML should consider multi-user aspects that can be extended to real-time collaborative 3D applications. Nevertheless, the World Wide Web architecture should be carefully considered as a base platform for real time collaborative 3D systems.

Several methods of providing 3D workspace awareness were discussed. In the shared 3D workspace, indicating accurate locations of camera viewports and locators is difficult and requires complex represenstions. 3D embodiments that indicate users' viewports and location can effectively show where other users are working. Object age features may also support workspace awareness by changing the object colour or representation over time. The incremental change of object age will provide some useful implicit awareness information, such as where other users have been working and how objects are created. Ways of representing workspace actions were also discussed.

Chapter 8

Conclusion

Design is viewed as a collective, collaborative and community process (Scrivener et al., 2000). Attempts have been made to support the design process using digital technologies. Most computer based design tools, however, are still single user oriented. There is a growing need to reflect the collective nature of design in the investigation of computer based design tools. The rapid development of communication and network technologies has resolved many technical constraints. User issues have become more critical in maximising the usefulness of computer based design tools in collaborative design environments.

This thesis has investigated the ways of providing an efficient design environment from a user perspective, focusing on a real-time collaborative 3D CAD system. The investigation adopted an evolutionary approach to the understanding of team design activities, identifying the issues of real-time collaborative design tools, constructing, developing and analytically evaluating a new framework and a prototype system.

This chapter concludes the thesis and has three parts. The objectives of the research set out in Chapter 1 are recalled and the findings are summarised by showing where and how these objectives were met in the thesis. Secondly, a summary of contributions that this research has made is presented. Finally, directions for future work are described based on the research carried out here. Final concluding remarks follow.

8.1 Progress on research objectives

The objectives of the research and the progress of each objective in the thesis are as follows.

i) To improve understanding of collaborative design activities and investigate the impact of collaborative technologies in collaborative design processes

This research objective was met by the preliminary experiments on team design processes (Chapter 3). Two laboratory-based experimental team design projects were observed and compared; one that was accomplished through a series of conventional face-to-face meetings and the other accomplished through a series of electronic meetings. Findings from the study indicated that team design processes proceeded systematically and the characteristics and roles of the meetings changed as the project progressed. Early meetings were more concerned with communication and information sharing, while later meetings were focused on collaborative productions and detail refinement of the design outcome. The designers in the computer-mediated situation experienced a number of technical problems with the collaborative tools, such as communication speed and limited screen resource. However, they could successfully complete the assignments without face-to-face contact. Comparison of the outcome shows the potentials of a successful design accomplishment with a well co-ordinated computer based collaborative design environment. Four types of collaborative activities were identified for the main areas of collaborative technology support: resource archiving and sharing, communication, process management, and co-working. Among these areas, a lack of support for real-time collaborative 3D design tools was identified although collaborative 3D activities are unique and important to designers who deal with 3D artefacts.

ii) To investigate an operational framework and a prototype system from a user's perspective, focusing on real-time collaborative 3D CAD

Issues to be addressed to develop a real-time collaborative 3D design tool were identified and the Shared Stage framework was proposed in the next phase of the research (Chapter 4 and Chapter 5). The Shared Stage was defined as a shared 3D workspace, in which collaborating designers share 3D objects in real-time. The framework emphasises the importance of the co-existence of individual and shared workspaces in complex 3D design activities. The Shared Stage module provide co-ordination and awareness support for the shared 3D workspace, while individual workspaces still support a conventional user interface method. The conceptual framework provided a basis for the implementation of prototype systems, which evolved through a series of development phases. In the first phase an existing commercial 3D CAD application was used as a host application and transformed to a collaborative 3D design tool. Although some features of the Shared Stage were examined by using a commercial application, a custom environment was required for the realisation of the Shared Stage framework. In the second phase of the prototype development, external software modules were developed as a plug-in application, which provided real-time collaborative features in an existing single-user 3D CAD application. Since the commercial 3D application still restricted the way that the Shared Stage module manages the data in a collaborative situation, a custom 3D CAD system, Syco3D, was developed considering means of extending real-time collaborative 3D features. General issues of real-time groupware applications, such as floor control, concurrency control and session management, have been considered as well as specific collaborative 3D design issues. The implementation involved the use of real-time application toolkits. By investigating a number of toolkits, a prototype development environment, supporting both real-time collaboration and 3D graphics, was customised and used as the main software platform for the realisation of the framework. The Shared Stage played a role in sharing visual representations and underlying structural information of the shared 3D workspace. The prototype real-time collaborative 3D CAD system, Syco3D, provided a number of collaborative features through two main elements of the Shared Stage: Synchronised Stage View and Data Structure Diagram. The Synchronised Stage View provided a synchronised 3D view, and various multi-user interface facilities, such as multiple cursors and a control lock for the synchronised camera view. The Data Structure Diagram provided a dynamic data exchange capability, status and underlying structural information about shared workspace. The sharing of representation and underlying structural information allowed designers to work together effectively and efficiently to accomplish sophisticated 3D design tasks. The Syco3D system addresses additional issues which the computer based design tools bring: smooth incorporation between individual and shared 3D workspace and generic 3D graphics support, and the support for workspace awareness. Further issues to be considered for the extension of real-time collaborative design to a higher level collaborative design environment were also discussed. These included 3D interaction techniques in the shared 3D workspace, the World Wide Web and VRML as a synchronous collaborative 3D design tool, 3D workspace awareness, and real-time collaborative 3D CAD tools for multi-disciplinary design teams.

iii) To evaluate and analyse the feasibility and the impact of the proposed system in a real-time collaborative design environment

This objective was met by usability evaluation using the prototype system. For a comparative usability experiment, a single user version of Syco3D system has been built. The designers accomplished experimental 3D CAD tasks collaboratively using the single-user version and collaborative version of the Syco3D system. Using the single user version, users mainly interacted by file exchanging, whereas using the collaborative version users could dynamically share all on-going workspace objects with other users. Various measures and the user response showed a positive impact to collaborative 3D modelling processes and a strong preference to the new environment. It was also found that the environment changes the way that people interact in the collaborative session. In particular, the way they interact in terms of verbal and non-verbal interaction. The quantitative analysis of user interaction in the usability evaluation indicated that the new environment improved the usability of 3D CAD systems in collaborative sessions. The exploratory results also showed that the shared stage extension was used mainly as a collaborative media, providing a platform for awareness support, co-ordination of ongoing models, and a medium for exchanging data.

8.2 Contributions

This research made several original contributions to design research.

- Primarily, issues were identified for the development of a real-time collaborative 3D CAD system. The issues related to 3D design tools were originally identified from a user's perspective.
- A new framework, called the Shared Stage, was demonstrated for real-time collaborative 3D design tools. Although previous work had investigated some real-time collaborative design tools to support real-time 2D drawing activities, little had been investigated in terms of real-time 3D collaborative design tools. It is expected that the Shared Stage can be used as an underlying structure for a new generation of computer based 3D design tools.
- The feasibility of the framework was illustrated through an analytic usability evaluation. The framework and the prototype successfully showed that the addition of the Shared Stage is a valuable addition to 3D design tools for real-time collaborative 3D tasks. In addition, the experimental evaluation showed the methodologies of analytical evaluation by employing various quantitative and exploratory analysis methods. The evaluation methods can be used for the evaluation of different computer based collaborative design tools.
- It was demonstrated that a new prototype development environment can be effectively used for real-time collaborative 3D applications. The software development environment combining real-time groupware and 3D graphics toolkits can become an effective prototyping environment for various synchronous multi-user 3D applications.
- The thesis increased the understanding of computer based collaborative design environments. The preliminary experimental study showed that a combination of currently available collaborative technologies is able to provide an environment in which distributed designers can complete a team design project without having face-to-face meetings. It also identified types of collaborative activities undertaken in computer based collaborative design environments and suggested collaborative tools based upon these kinds of activities.

- An original 3D modelling tool was designed and built. A number of collaborative features supporting collaborative 3D interactions and workspace awareness were incorporated. A new set of collaborative technologies was investigated for distributed collaborative design activities. This 3D modelling tool may be used for actual design practice of simple geometric modelling or for investigating other 3D CAD issues.

The ideas and findings of this thesis have been published in referred conference and journal articles (Nam and Wright, 1997; Nam and Wright, 1998; Nam, 1998; Nam and Wright, 2000, Nam and Wright, 2001).

8.3 Future work

This research raised many new questions as outlined below. General areas for future research are described based on what have been explored so far.

- *Incorporation of inter-communication tools:* Effective ways of providing interpersonal communication support is important when design teams engage in collaborative 3D design activities. For distributed collaboration, video conferencing with shared 2D whiteboard, shared database systems and synchronous and asynchronous collaborative applications should be smoothly integrated. Further investigation should be carried out to integrate computer-mediated communication tools in a real-time collaborative 3D design tool.
- *3D data recording support in a real-time collaborative design tool*: There are many aspects to be improved in terms of the usability of the prototype system. One of the features to be further investigated is that of providing effective recording of data created by multiple people. Current systems lack support for recording the result of collaborative 3D modelling. Efficient ways of recording the workspace under multi-user environments is significantly different from single use. A new specification and strategies to manage multi-user oriented databases should be investigated further.
- *Extending collaborative 3D systems for more than two users:* This research focused on a situation where two designers work together for the same 3D CAD design

project. Technically more than two people are able to use our system. However, considerations should be given to the evaluation of interactions between more than two people using the system. Further investigation can be conducted to examine the feasibility of the Shared Stage framework in a medium or large size design team.

- *Further investigation of 3D workspace awareness*: In the collaborative situation, tools to provide information about activities of other users and workspaces help designers to understand what they have done, what they are doing and what they will be doing. Some techniques were discussed but further investigation is required to provide these unique aspects of 3D workspace awareness. A strategy is required to deliver appropriate 3D awareness information and identify awareness support for collaborative 3D CAD tasks for the future generation of collaborative 3D CAD systems.
- *3D interaction techniques in collaborative design environments:* Comparative investigations of 3D interaction techniques between single-user and multi-user design tools could be carried out. 3D interaction techniques were briefly discussed for real-time collaborative 3D design tools. Based upon the investigation of 3D interaction techniques in single-user design tools, it is also necessary to explore ways of providing the most appropriate 3D interaction in collaborative situations. Special requirements for 3D interaction techniques in a collaborative situation should be further explored.
- *Multi-disciplinary collaboration support:* The framework of the thesis focused on collaborative 3D activities normally taking place in a single-disciplinary team. 3D collaboration can take place between professionals from different domains, such as product planning, design, development, manufacturing and marketing. 3D collaborative CAD tools should also provide effective communication for such a multi-disciplinary collaboration. It might be necessary to control the information type depending on the current collaboration. The shared 3D workspace can be modified for multi-disciplinary collaboration and further investigation should be conducted to extend the Shared Stage framework to a multi-disciplinary collaboration.
- *Employing the World Wide Web*: The World Wide Web is an attractive platform for collaborative application. Although current Web architecture is stateless and difficulties arise when providing synchronous collaboration features, further

consideration might be given to ways of incorporating synchronous 3D collaboration features within the Web structure.

- *Further user studies*: New experiments can be designed in order to investigate further impacts of real-time collaborative features. In order to analyse significant interactions between variables using quantitative analysis methods, the nature of experimentation tasks and variables needs to be carefully controlled. User studies can also be carried out in a real-world setting. The reliability and functionality of the prototype system should be improved to address the complexity of actual 3D CAD design practices of designers.

8.4 Conclusion

This research investigated a collaborative design environment focusing on a real-time 3D collaborative CAD system providing the possibility for supporting collaborating designers involved in 3D design activities. Although many issues still need to be addressed in the application of a successful computer based collaborative design environment, this research should provide some design criteria for the next generation of computer based design tools. Further work is suggested in several different areas; these include the incorporation of the system into a collaborative design environment, the enhancement of the system features, an investigation of collaborative 3D interaction techniques, and further user evaluation.

REFERENCES

[1] ActiveWorlds (2000), Available at http://www.activeworlds.com

[2] Akin, O. and Lin, C. (1996), "Design protocol data and Novel Design Decisions", in Cross, N. Christiaans, H. and Dorst, K. (eds.) *Analyzing Design Activity*, Wiley, Chichester, pp. 35-64.

[3] Alexandar, C. (1964), *Notes on the Synthesis of Form*, Cambridge MA Harvard University Press.

[4] Alias|Wavefront (2000), *Alias Manual Version 9.5* Alias/Wavefront, a division of Silicon Graphics Limited.

[5] Anupam, V. & Bajaj, C.L. (1994), "Shastra: Multimedia Collaborative Design Environment", *IEEE Multimedia*, Summer 1994, pp. 39-49.

[6] Arthur, K., Booth, K. and Ware, C. (1993), "Evaluating 3D task performance for fish tank virtual worlds", ACM *Transactions on Information Systems* Vol. 11, No. 3, pp. 239-265.

[7] Asimow, M. (1962), *Introduction to Design*, Englewood Cliffs, NJ, Prentice-Hall.

[8] Baeker, R. M. (eds.) (1993), *Readings in Groupware and Computer Supported Cooperative Work: Assisting Human-Human Collaboration*, Morgan Kaufmann Publishers Inc., San Mateo, CA.

[9] Baecker, R. M., Glass, G., Mitchell, A. and Posner, I. R. (1994), "SASSE: the Collaborative Editor", a video tape presented at the 1994 ACM *Conference on Human Factors in Computing Systems*, May 1994.

[10] Baecker, R. M., Nastos, D., Posner, I.R. and Mawby K.L. (1993), "The User-centred Iterative Design Of Collaborative Writing Software", in Baecker, R. M.,

Grudin, J., Buxton, W. and Greenberg, S. (eds.) *Human-Computer Interaction: Toward the Year 2000*, pp. 775-782.

[11] Bajura, N., Fuchs, H., Ohbuchi, R. and Merging T. (1992), "Merging Virtual Objects with the Real World: Seeing Ultrasound Imagery Within the Patient", In Proceedings of *SIGGRAPH'92*, New York ACM Press, pp. 203-210.

[12] Bannon, L. (1997) "CSCW - A challenge to certain (G)DSS perspectives on the role of decisions, information, and technology in organizations?" In Humphreys, P., Ayestaran, P., McCosh, A. and Mayon-White B. (eds.) (1997) *Decision Support in Orgamizational Transformation.* London: Chapman

[13] Bayaa, V. and Leifer, L. (1996), "Understanding Information Management in Conceptual Design", in Cross, N., Christiaans, H. and Dorst, K. (eds.) *Analyzing Design Activity*, pp.151-168.

[14] Bentley, R., Busbach, U. and Sikkel, K. (1996), "The Architecture of the BSCW Shared Workspace System", Busbach, U., Kerr, D. and Sikkel, K. (eds.) *CSCW and the Web - Proceedings of the 5th ERCIM/W4G Workshop.* Arbeitspapiere der GMD 984, GMD, Sankt Augustin.

[15] Billinghurst, M. and Kato, H. (1999), "Collaborative Mixed Reality", in *Proceedings of the First international symposium on Mixed Reality*, pp 261-284. Berlin: Springer Verlag.

[16] Birman, K. P. and Joseph, T. A. (1987), "Reliable communication in the presence of failures", *ACM Transactions on Computer Systems*, February 1987, pp. 47-76.

[17] Bly, S. and Minneman, S. (1990), "Commune: A Shared Drawing Surface", in *Procedings of ACM Conference on Office Information Systems COIS'90,* Boston, Mass., pp. 185-192.

[18] Brink, T. (1998), *Groupware design issues*, Usability First Home Page, Available at http://www.usabilityfirst.com/groupware/design-issues.html.

[19] Brink, T. and Gomez, L. (1992), "A Collaborative Medium for the Support of Conversational Props", in Proceedings of *ACM Conference on Computer-Supported Cooperative Work CSCW'92,* Toronto, Canada, pp. 171-178.

[20] Chang, J. M. and Maxemchuck, N. F. (1984), "Reliable broadcast protocols". *ACM Transactions on Computer Systems*, Vol. 2, No. 3, pp. 251-273.

[21] Chabert, A., Jackson, E. and Pietrovicz, S. (1997), "NCSA Habanero - Synchronous collaborative framework and environment", *ECSCW97 conference supplement*, Lancaster, UK, pp. 7-8.

[22] CoCreate Software (1999), *OneSpace: A Collaborative Design Infrastructure, A CIMData Product Review*, Available at http://www.cocreate.com.

[23] Cross, N., Dorst, K. and Roozenburg N. (eds.) (1992), *Research in Design Thinking*, Delft University Press, Delft, Netherlands.

[24] Cross, N., Christiaans, H. and Dorst, K. (eds.) (1996), *Analyzing Design Activity,* Wiley, Chichester.

[25] Cross, N. and Cross, A. C. (1996), "Observation of teamwork and social process in design", in Cross, N., Christiaans H. and Dorst, K. (eds.) *Analyzing Design Activity*, pp. 291-317.

[26] Cruz-Neira, C., Sandin, D.J., Defanti, T.A., Kentyon, R.V. and Hart, J.C. (1992), "The CAVE: Audio Visual Experience Automatic Virtual Environment", *Communication of the ACM*, Vol. 35, No. 6, pp. 65-72.

[27] De Boo, M. (1999), "Creative design using gestures: New Technologies for three-dimensional design work", *Delft Outlook*, April, 1999, pp. 3-8.

[28] Deering, M. (1992), "High resolution virtual reality", *Computer Graphics* (ACM), Vol. 26, No. 2, pp. 195-202.

[29] Dix, A. (1997), "Challenges for Cooperative Work on the Web: An analytic approach", *Computer Supported Cooperative Work: The Journal of*

Collaborative Computing: Special Issue on CSCW and the Web, Kluwer, Vol. 6, No. 2-3, pp. 135-156.

[30] Ellis, C. A., Gibbs, S. J. and Rein, G. L. (1991), "Groupware: Some issues and experiences", *Communication of the ACM*, Vol. 34, No. 1, pp. 38-58.

[31] Engeli M. and Mueller A. (1999), "Digital Environments for Collaboration - Architecture, Communication, Creativity", *in proceedings of: ACADIA 1999 conference*, Salt Lake City, Utah, October 28-31, 1999.

[32] Feiner, S., MacInyre, B. and Seligmann, D. (1993), "Knowledge-Based Argumented Reality", *Communication of the ACM*, Vol. 36, No. 7, pp. 53-62.

[33] Gisi M. A. and Sacchi, C. (1994), "Co-CAD: A Collaborative mechanical CAD system", *PRESENCE*, Vol. 3(Fall), pp. 341-350.

[34] Goldschmidt, G. (1995), "The designer as a team of one", *Design Studies*, Vol. 16, pp. 189-209.

[35] Graham, T.C.N., Urnes, T. and Nejabi, R. (1996), "Efficient distributed implementation of semi-replicated synchronous groupware", In *Proceedings of the ACM UIST'96 User Interface Software and Technology*, Seattle, Washington, November 6-8, 1996, pp. 1-10.

[36] Greco, J. (2000) "Collaboration for the MCAD e-World", CADENCE Magazine, Published by Kelly Schulman, USA, Features, September, 2000

[37] Greenberg, S., Hayne, S. and Rada, R. (eds.) (1995), *Groupware for Real-Time Drawing: A Designer's Guide*, McGraw-Hill Book Company.

[38] Greenberg, S. and Roseman, M. (1999), "Groupware Toolkits for Synchronous Work", in Beaudouin-Lafon M. (eds.) *Computer Supported Co-operative Work: Trends in Software*, John wiley & Sons, England, pp. 135-168.

[39] Greenberg, S., Roseman, M., Webster, D. and Bohnet, R. (1995), "Human and technical factors of distributed group drawing tools", Greenberg, S., Hayne, S.

and Rada, R. (eds.) *Groupware for Real-Time Drawing : A Designer's Guide*, McGraw-Hill Book Company, pp. 37-62.

[40] Greenhalgh, C.M. and Benford, S.D. (1995), "MASSIVE: A Collaborative Virtual Environment for Tele-conferencing", *ACM Transactions on Computer Human Interfaces (TOCHI)*, Vol. 2, No. 3, pp. 239-261.

[41] Greif, I., Seliger, R. and Weihl, W. (1992), "A Case Study of CES: A Distributed Collaborative Editing System Implemented in Argus", In *IEEE Transactions on Software Engineering*, Vol. 18, No. 9, pp. 827-839.

[42] Gribnau, M. W. (1999) *Two-Handed Interaction in Computer Supported 3D Conceptual Modelling*, PhD Thesis, Delft University of Technology, Delft, The Netherlands, ISBN 90-9013038-1

[43] Grudin, J. (1994), "Computer-Supported Co-operative Work: Its History and Participation", *IEEE Computer*, Vol. 27, No. 5, pp. 19-26.

[44] Grudin, J. (1994a), "Groupware and Social Dynamics: Eight Challenges for Developers", *Communication of the ACM*, Vol. 37, No. 1, pp. 92-105.

[45] Gutwin, C. (1997), *Workspace Awareness in Real-Time Distributed Groupware*, PhD Thesis, The University of Calgary.

[46] Gutwin, C. and Greenberg, S. (1998), "Effects of Awareness Support on Groupware Usability", *Proceedings of CHI'98: Human Factors in Computing Systems*, pp. 511-518.

[47] Gutwin, C., Greenberg, S., and Roseman, M. (1996), "Workspace Awareness in Real-Time Distributed Groupware: Framework, Widgets, and Evaluation", *PEOPLE AND COMPUTERS*, 1996, No. EDIT 11, pp. 281-298.

[48] Harper, R. and Hughes, J. (1993), "What A F-ing System! Send'em all to the same place and then expect us to stop 'em hitting: Making Technology Work in Air Traffic Control", in Button, G. (ed), *Technology in Working Order*, Routedge, London, pp. 127-144.

[49] Harrison, S. and Minneman, S. (1996), "A Bike in Hand: a Study of 3D Objects in Design", in Cross, N., Christiaans, H. and Dorst, K. (eds.) *Analyzing Design Activity*, pp. 417-436.

[50] Haymaker, J., Keel, P., Ackermann, E. and Porter W. (2000), "Filter mediated design: generating coherence in collaborative design", *Design Studies,* Vol. 21, No. 2, pp. 205-220.

[51] Hewlett Packard Company (2000), "Virtual Prototyping: The future arrives in the form of a Showcase Automobile", Available at http://www.hp.com/visualize/demos/supercar/predwp.pdf.

[52] Hill, R.D. (1992), "The abstraction-Link-View paradigm: Using constraints to connect user interfaces to applications", In *Proceedings of the ACM SIGCHI'92 Conference on Human Factors In Computing Systems*, Monterey, California, May 3-7, 1992, pp. 335-342.

[53] Hill, R. D., Brink, T., Rohall, S.L., Patterson, J.F. and Wilner, W. (1994), "The Rendezvous architecture and language for constructing multi-user applications", *ACM Transactions on Computer Human Interaction*, Vol. 1, No. 2, pp. 81-125.

[54] Hollan, J., and Stornetta, S. (1992), Beyond being there. *Proceedings of the ACM CHI'92 Conference on Computer-Human Interaction* (Monterey, CA, 3-7 May 1992), pp. 119-125.

[55] Houde, S. (1992) "Iterative design of an interface for easy 3-D direct manipulation", *Proceedings of CHI'92*, May 3-9 1992, pp. 135-141.

[56] Hudak, P. and Wadler, P. (1991), *Report on the functional programming language Haskell (v1.1)*, Technical Report YALEU/DCS/RR777, Yale University

[57] Ishii, H. (1990), "TramWorkStation: towards a seamless shared workspace", in *Proceedings of CSCW '90*, ACM Press, New York, pp. 13-26.

[58] Ishii, H., Arita, K. and Yagi, T. (1993), "Beyond videophones: TramWorkstation-2 for narrowband ISDN", in *Proceedings of ECSCW'93*, Kluwer Academic Press, Dordrecht, pp. 333-346.

[59] Ishii. H., Kobayashi, M. and Grudin, J. (1992), "Clearboard: A Seamless Media for Shared Drawing and Conversation with Eye-Contact", In *Proceedings of CHI'92,* ACM Press, pp. 525-532.

[60] Ishii, H., Kobaysshi, M. and Grudin, J. (1995). "Integration of Interpersonal Space and Shared Workspace: Clearboard Design and Experiments", in Greenberg, S., Hayne, S. and Rada, R. (eds.) *Groupware for Real-time drawing: A Designer' Guide*, pp. 96-125.

[61] ISO (1998), *Guidance on usability*, ISO 9241 Part 11.

[62] Jasnoch, U., Kress, H., Schroeder, K., Ungerer, M. (1994), "CoConut: Computer Support for Concurrent Design using STEP", in Proceedings of 3rd IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Interprises (WET-ICE) April 17-19, IEEE Computer Society Press

[63] Johansen, R. (1988), *Groupware: Computer support for business teams.* NY: The Free Press.

[64] Jones, C. (1970), *Design Methods: seeds. of human futures*, Chichester: John Wiley & Sons.

[65] Kao, Y. C. and Lin, G. C. (1996), "CAD/CAM collaboration and remote machining", *Computer Integrated Manufacturing Systems*, Vol. 9, No. 3, pp. 149-160.

[66] Karsenty, A. and Beaudouin-Lafon, M. (1993), "An algorithm for distributed groupware application", *Proceedings of the 13th International Conference on Distributed Computing Systems*, IEEEE Press, pp. 195-202.

[67] Koch, T. and Appelt, W. (1998), "Beyond Web Technology - Lessons learnt from BSCW", In Society, I. C. (ed.), *Proceedings of the Seventh IEEE*

International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE '98), 17-19 June 1998, Stanford University, CA: IEEE Computer Society, pp. 176-181.

[68] Krasner, G. E. and Pope, S. T. (1988), "A cookbook for using the model-view-cntroller user interface paradigm in Smalltalk-80", *Journal of Object Oriented Programming,* Vol. 1, No. 3, August/September 1988, pp. 26-49.

[69] Kraut, R. E., Egido, C. and Galegher, J. (1990), "Patterns of contact and communication in scientific research collaborations", in *Intellectual Teamwork: Social Foundation of Cooperative Work*, Lawrence Erlbaum Associates, New York, pp. 149-72.

[70] Kuttner, B.C. (1997), "CAD Conferencing", C-TAD Systems Inc., Available at http://www.ctad.com.

[71] Leland, M. D. P., Fish, R. S. and Kraut, R. E. (1988), "Collaborative Document Production Using Quilt", in *Proceedings of the Conference on Computer-Supported Cooperative Work 1988,* New York: ACM Press, pp. 206-215.

[72] Lotus (2000), "Lotus Notes", Available at http://www.lotus.com.

[73] Maher, M. L., Simoff, S. J. and Cicognani, A. (1997), "Observations from an experimental study of computer-mediated collaborative design" in Maher, M. L., Gero J. S. and Sudweeks, F. (eds.) *Formal Aspects of Collaborative CAD*, University of Sydney, pp. 165-186.

[74] Maher, M. L., Cicognani, A. and Simoff, S. (1998), "An experimental study of computer mediated collaborative design", *International Journal of Design Computing,* Available at http://www.arch.usyd.edu.au/kcdc/journal.

[75] Maher, M., Simoff, S. and Cicognani, A. (1999), *Understanding Virtual Design Studios*, Springer

[76] Mazijoglou, M., Scrivener, S. and Clark, S. (1995), "Representing Design Workspace Activity", in Cross, N., Christiaans H. and Dorst, K. (eds.) *Analyzing Design Activity*, pp. 389-415.

[77] McCarthy, J.C., Miles, V.C. and Monk, A.F. (1991), "An experimental study of common ground in text-based communication", in Robertson, S.P., Olson, G.M. & Olsen J.S. (eds.), *Human Factors in Computing Systems: CHI'91 Conference Proceedings* (ACM: New York), pp. 181-193.

[78] McCarthy, J. C. and Monk A. F. (1994), "Measuring the quality of computer-mediated communication", *Behaviour & Information Technology*, Vol. 13, No. 5, pp. 311-319.

[79] McGuffin, L. and Olson, G.M. (1992), *ShrEdit, a Shared Electronic Workspace*, Cognitive Science and Machine Intelligence Laboratory, University of Michigan.

[80] Milgram, P. and Kishino, F. A. (1994), "Taxonomy of mixed reality visual displays", *IEICE Transactions on information and Systems, Special issue on Networked Reality*, Dec. 1994, pp. 1321-1329.

[81] Minneman, S.L. (1991), *The social construction of a technical reality: empirical studies of group engineering design practice*, Xerox PARC Technical Report (Doctoral Dissertation, Stanford University)

[82] Minneman, S.L. and Bly, S.A. (1991), "Managing a Trois: a Study of a Multi-user Drawing Tool in Distributed Design Work", In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI'91)*, New Orleans, Louisiana, April 27 - May 2, 1991, Addison-Wesley, Reading, MA, 1991. pp. 217-224.

[83] Moran, T. P., McCall K., Melle, B., Pedersen, E. R. and Halasz F. G. (1995), "Some Design principles for Sharing in Tivoli, a Whiteboard Meeting-Support Tool", in Greenberg, S., Hayne, S. and Rada, R. (eds.) (1995), *Groupware for*

Real-Time Drawing: A Designer's Guide, McGraw-Hill Book Company, pp. 24-36.

[84] Nam, T-J. (1997) "Investigation into Collaborative Design Environment: Experimental Study on Team Design Activities in a Face-to-Face and a Computer-mediated environment", *Case History ICSID '97 The Humane Village Congress*, 23 – 27 August 1997, Toronto, Canada

[85] Nam, T-J. (1998) "An investigation of multi-user design tools for collaborative 3D modelling", *Proceedings of CSCW98 (Computer Supported Cooperative Working) ACM*, Seattle, USA, p. 420 [Doctoral Colloquium]

[86] Nam, T-J. and Wright, D. (1998), "CollIDE: A Shared 3D Workspace for CAD", *Proceedings of 4th International Conference on Networking Entities NETIES'98: Networking for the Millennium,* 15 - 16 October 1998, Leeds., UK, pp. 103-105.

[87] Nam, T-J and Wright, D. (2000) "Observations of Team Design Processes and the Impact of Collaborative Technologies", *Adjunct Proceedings of CoDesigning 2000*, 11-13 September, 2000, Coventry, UK, pp. 131 -136

[88] Nam, T-J. and Wright D. (2001) "Syco3D: A Real-time Collaborative 3D CAD system", *Design Studies* (in press)

[89] Neuwirth, C.M., Kaufer, D., Chandhok, R. and Morris, J. (1994), "Computer Support for Distributed Collaborative Writing: Defining Parameters of Interaction", *CSCW `94: Proceedings of the Conference on Computer-Supported Cooperative Work.* pp. 145-152.

[90] Norman, D. (1998), *The invisible computer*, The MIT Press, Cambridge, Massachusetts

[91] Nunamaker, J. F., Dennis, A. R. , Valacich, J. S., Vogel, D. R. and George, J. F. (1991), "Electronic meeting systems to support group work", *Communications of the ACM*, Vol. 34, No. 7, pp. 40-61.

[92] Olson, G. M., Olson, J. S., Carter, M. R. and Storrosten, M. (1992), "Small Group Design Meetings: An Analysis of Collaboration", *HUMAN-COMPUTER INTERACTION*, Vol. 7, pp. 347-374.

[93] Ousterhout, J.K. (1994), *Tcl and the Tk Toolkit*, Addison-Wesley Publishing Company.

[94] Pacull, F., Sandoz, A., and Schiper, A. (1994), "Duplex: A Distributed Collaborative Editing Environment in Large Scale", In *CSCW '94: Proceedings of the Conference on Computer-Supported Cooperative Work*, New York: ACM Press, pp. 165-173.

[95] Pang, A. and Wittenbrink, C.M. (1997), "Collaborative 3D Visualization with Cspray", *IEEE Computer Graphics and Applications*, Vol. 17, No. 2, March 1997, pp. 32-41.

[96] Patterson, J. F., Day, M. and Kucan, J., (1996), "Notification servers for synchronous groupware", In *proceedings of the ACM CSCW'96 Conference on Computer Supported Cooperative Work*, Bostonn, Mass., November 16-20, 1996, pp. 122-129.

[97] Polhemus Inc. (2000), "3SPACE FASTRAK", Available at http://www.polhemus.com/home.htm.

[98] Prakash, A. (1999), "Group Editors", in Beaudouin-Lafon M. (eds.) *Computer Supported Co-operative Work: Trends in Software*, John Wiley & Sons, England, pp. 103-134.

[99] Prakash, A. and Knister, M., (1992), "Undoing actions in collaborative work", In *Proceedings of the Fourth ACM Conference on Computer-Supported Cooperative Work*, Toronto Canada, October 1992, pp. 273-280.

[100] Prakash, A. and Shim, H. (1994), "DistView: Support for building efficient collaborative aplications using replicated objects", In *Proceedings of the Fifth ACM Conference on Computer-Supported Cooperative Work*, Toronto Canada, October 1994. ACM Press, pp. 153-164.

[101] Pugh, S. (1996), *Creating innovative products using total design: the living legacy of Stuart Pugh*, Addison-Wesley.

[102] RealityWave (2000), Available at http://www.realitywave.com

[103] Robson, C. (1996), *Experiment, Design and Statistics in Psychology*, Penguin Psychology.

[104] Roseman, M. and Greenberg, S. (1996), "Bruilding Real-Time Goupware with Groupkit, A Groupware Toolkit", *ACM Transactions on Computer-Human Interaction*, March 1996, Vol. 3, No. 1, pp. 66-106.

[105] Roseman, M. and Greenberg, S. (1996a), "TeamRooms: Network Places for Collaboration", *Proceedings of ACM CSCW '96*, pp. 325 – 333.

[106] Rosenman, M.A. and Gero, J.S. (1996), "Modelling multiple views of design objects in a collaborative CAD environment", *Computer-Aided Design*, Vol. 28, No. 3, pp. 193-205.

[107] Rosenman, M. A. & Gero, J. S. (1997), "Collaborative CAD Modelling in Multidisciplinary Design Domains", Available at http://www.arch.usyd.edu.au/~mike/PAPERS/collabCAD.html.

[108] Saad, M. & Maher, M.L. (1996), "Shared understanding in computer-supported collaborative design", *Computer-Aided Design*, Vol. 28, No. 3, pp. 183-192.

[109] Saad, M. (1994), *Shared Understanding in Synchronous Collaborative Design*, PhD Thesis, Department of Architectural and Design Science, University of Sydney.

[110] Sachs, E., Roberts, A. and Stoops, D. (1991) "3-Draw: A tool for designing 3D shapes", *IEEE Computer Graphics & Applications*, Vol. 11, November 1991, pp. 18-26

[111] Sakai, A. (1996). “Flying Fingers: A tool for three-dimensional shared workspace”, *Conference on Human Factors in Computing Systems Proceedings (CHI’96)*, pp. 295-296.

[112] Schmitt G., Stouffs R. and Engeli M. (1997), “An architecture for collaborative design support: Models, tools and interfaces”, in M.L. Maher, J.S. Gero, F. Sudweeks (eds.), *Proceedings of IFIP WG 5.2 Workshop on FormalAspects of Collaborative CAD*, Key Center of Design Computing, University of Sydney, pp. 125-142.

[113] Schroeder, W., Martin, K. and Lorensen, B. (1996), *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*, Prentice Hall PTR.

[114] Schuckmann, C., Kirchner, L., Schummer, J. and Haake, J.M. (1996), “Designing object-oriented synchronous groupware with COAST”, In *Proceedings of the ACM Conference on Computer-Supported Co-operative Work (CSCW96)*, pp. 30-38.

[115] Scrivener, S. A. R., Harris, D., Clark, S. M., Rockoff, T. and Smyth, M. (1993), “Designing at a distance via real-time designer-to-designer interaction”, *Design Studies*, Vol. 14, No. 3, pp. 261-282.

[116] Shu, L. and Flowers, W. (1992), “Groupware Experiences in Three-Dimensional Computer-Aided Design”, *Proceedings of CSCW’92*, pp. 179-186.

[117] Sigma Design International (2000), Available at http://www.ezmeeting.com

[118] Silicon Graphics (2000), “IRIX Annotator”, Available at http://www.sgi.com/Products/Annotator.html

[119] Simon, H.A. (1981), *The Sciences of the Artificial*, The second edition, MIT Press, Cambridge, MA.

[120] Sproull, L. and Kiesler, S. (1991), "A Less in Electronic Mail", *Connections: New Ways of Working in the Networked Organization*, The MIT Press, pp. 177-184.

[121] Stefik, M. M., Foster, G., Bobrow, D., Kahn, K., Lanning, S. and Suchman, L. (1987a), "Beyond the Chalkboard: Computer Support for Collaboration and Problem-Solving in Meetings", *Communications of the ACM*, Vol. 30, No. 1, pp. 32-47.

[122] Stefik, M., Bobrow, D. G., Foster, G., Lanning, S., Tatar, D. (1987b) "WYSIWIS revised: Early Experiences with Multi-user Interfaces", *ACM Transactions of Office Information Systems*, Vol. 5, No. 2, pp. 147-167.

[123] Stenius, M. (1996) *Collabroative object modelling in virtual environments*, Master's thesis, School of Computer Science and Engineering, Royal Institute of Technmology, KTH, Sweden

[124] Strauss, A.L. (1987), *Qualitative Analysis for Social Scientists*, Cambridge University Press, Cambridge.

[125] Streitz, M.A., Geibler, J., Haake, J.M. and Hol J. (1994), "DOLPHIN: Integrated Meeting Support Across Local and Remote Desktop Environments and Liveboards", In *CSCW '94: Proceedings of the Conference on Computer-Supported Cooperative Work*. New York: ACM Press, pp. 345-358.

[126] Suchman. L, (1987), *Plans and Situated Action: The problem of human-machine communication*, Cambridge University Press, Cambridge, UK.

[127] Tang, J. C. (1989), *Listing, Drawing, and Gesturing in Design: A Study of the Use of Shared Workspaces by Design Teams*, Xerox PARC Technical Report (Doctoral dissertation, Stanford university)

[128] Tang, J. C. (1991), "Findings from observational studies of collaborative work", *Man-Machine Studies,* Vol. 34, pp. 143-160.

[129] Tang, J.C. and Leifer, L. (1988), "A Framework for Understanding the Workspace Activity of Design Teams", *Proceedings of the Second Conference on Computer-Supported Cooperative Work,* pp. 244- 248.

[130] Tang, J. and Leifer, L. (1991), "An observation methodology for studying group design activity", *Research in Engineering Design*, Vol. 2, No. 4, pp. 209-220.

[131] Tang,J.C. and Minneman, S. L. (1991), "VideoDraw: A Video Interface for Collaborative Drawing", *ACM Transactions on Information Systems,* Vol. 9, No. 2, pp. 170-184.

[132] Teamwave Software Ltd (2000), "Teamwave workplace", Available at http://www.teamwave.com (Accessed: March 2000)

[133] Tec (1993), *Xshare Version 0.9 for the X windows system*, ETI

[134] Toye, G. Gutkosky, M., Leifer L, Tenenbaum, J. and Gilcksman, J (1993), "SHARE: A Methodology and Environment for Collaborative Product Development", in *Post-Proceedings of the IEEE Infrastructure for Collaborative Enterprises*, pp. 1-16.

[135] Trever, J., Koch, T. and Woetzel, G. (1997), "MetaWeb: Bringing synchronous groupware to the World Wide Web", in Hughes, J. et al. (eds.), *Proceedings of the Fifth European Conference on Computer Supported Cooperative Work*, pp. 65-80.

[136] Turoff M. (1991), "Computer Mediated Communication Requirements for Group Support", *Journal of Organizational Computing*, Vol. 1, pp. 85-113.

[137] Venolia, D. (1993), "Facile 3D direct manipulation", in *INTERCHI'93 Conference Proceedings*, pp. 31-36.

[138] Welch, B.B. (1995), *Practical programming in Tcl and TK*, Prentice-Hall PTR.

[139] Wilson, P. (1991), *Computer Supported Co-operative Work: An Introduction*, Oxford, Intellect Books.

[140] Winograd, T. (1987), "A Language/Action Perspective On the Design of Co-operative Work", *Human Computer Interaction*, Vol. 3, No. 1, pp. 3-30.

[141] Wojitowicz, J. (ed.) (1995), *Virtual Design Studio*, Hong Kong University Press, Hongkong.

[142] Yee, S., Mitchell, W. Naka, R., Morozumi, M. and Yamaguchi, S. (1998), "The Kumamoto-Kyoto-MIT Collaborative Project: A Case Study on the Design Studio of the Future", In *proceedings of CoBuild'98*, pp. 80-93.

[143] Zeleznik, B. Holden, L., Capps, M., Abrams, H. and Miller T. (2000), "Scene-Graph-As-Bus: Collaboration between Heterogeneous Stand-alone 3-D Graphical Applications", *EUROGRAPHICS* August, 2000, Vol. 19, No. 3

Appendix A: Results of the Preliminary Experimental Study

A.1 The outcome of the distributed team design project with a series of face-to-face meetings

A.1.1 Torch Specification

Torch specification

Solar Powercharger

- Solar powered charging unit
- Holds one cell
- Battery status indicator
- Intelligent recharge system:
  surplus energy stored in the energy reservoir
  Recycled to recharge flat battery
- Energy reservoir unit (1/3 cell capacity)

Torch Unit

- Dual filament bulb
  High intensity
  Wide beam
- Cellular battery (size 100mm x 15mm)
- Emergency light (using standard flasher unit)
- Multi-function switch:
  Spot light
  Wide beam
  Emergency light

Usage

Battery charger is placed at the back of the dashboard and is fixed (temporarily) on to the windscreen, using two rubber suckers. This allows the charger's solar panels to absorb maximum solar energy. This energy is harnessed to charge the spare cell.

When a new cell is introduced into the charger the unit checks the cell's charge. If the cell is not fully discharged, it will discharge the remaining energy into the 'Energy Reservoir Unit'. This is then recycled back into the cell once it has fully dissipated [This is required as the cell must be fully discharged before charging, else damage may be incurred].

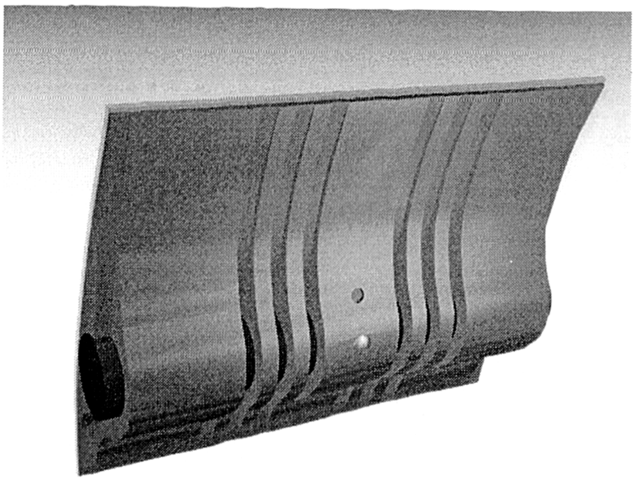
When the charger LED turns from red to green, this indicates that the cell is fully charged. The charger will stop charging the cell once the cell has obtained full charge.

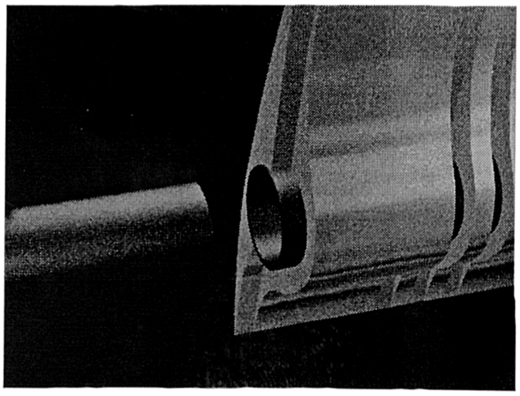
The torch has a muti-functional switch. This is used to:

Toggle between full and wide beam
Switch on the flashing emergency signal.

Cell is housed in the main body of the torch and can be simply accessed by unscrewing the back end. The head of the torch unscrews, to enable easy bulb replacement.

A.1.2 Presentation Drawings





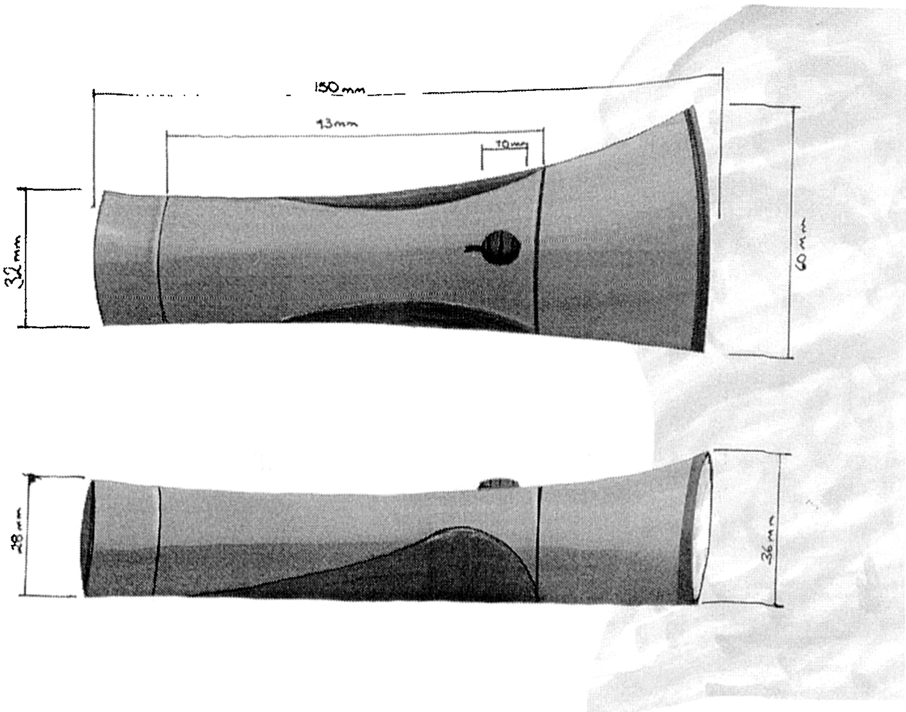
Top left: Torch

Top right: Solar Battary Charger

Lower left: the assembly of the charger and a battery

Bottom: Dimensioned Drawing

Torch Dimension



A.2 The outcome of the distributed team design project with a series of computer-mediated meetings

A.2.1 Specification

JEDI
Design Corp.

Car torch

Use and purpose

This car torch is a robust but stylish design and can be used as a repair light, emergency warning light and also as a general lighting device. There is a powerful magnet for attaching the torch to the car body.

Materials used are light-weight, durable and easy to use in modern manufacturing processes.

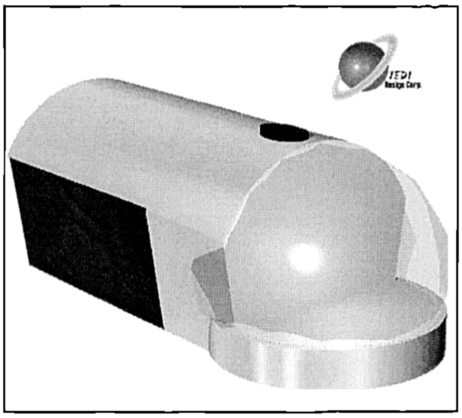
Battery is using standard NiCd cells packed in a durable and easily detachable casing.

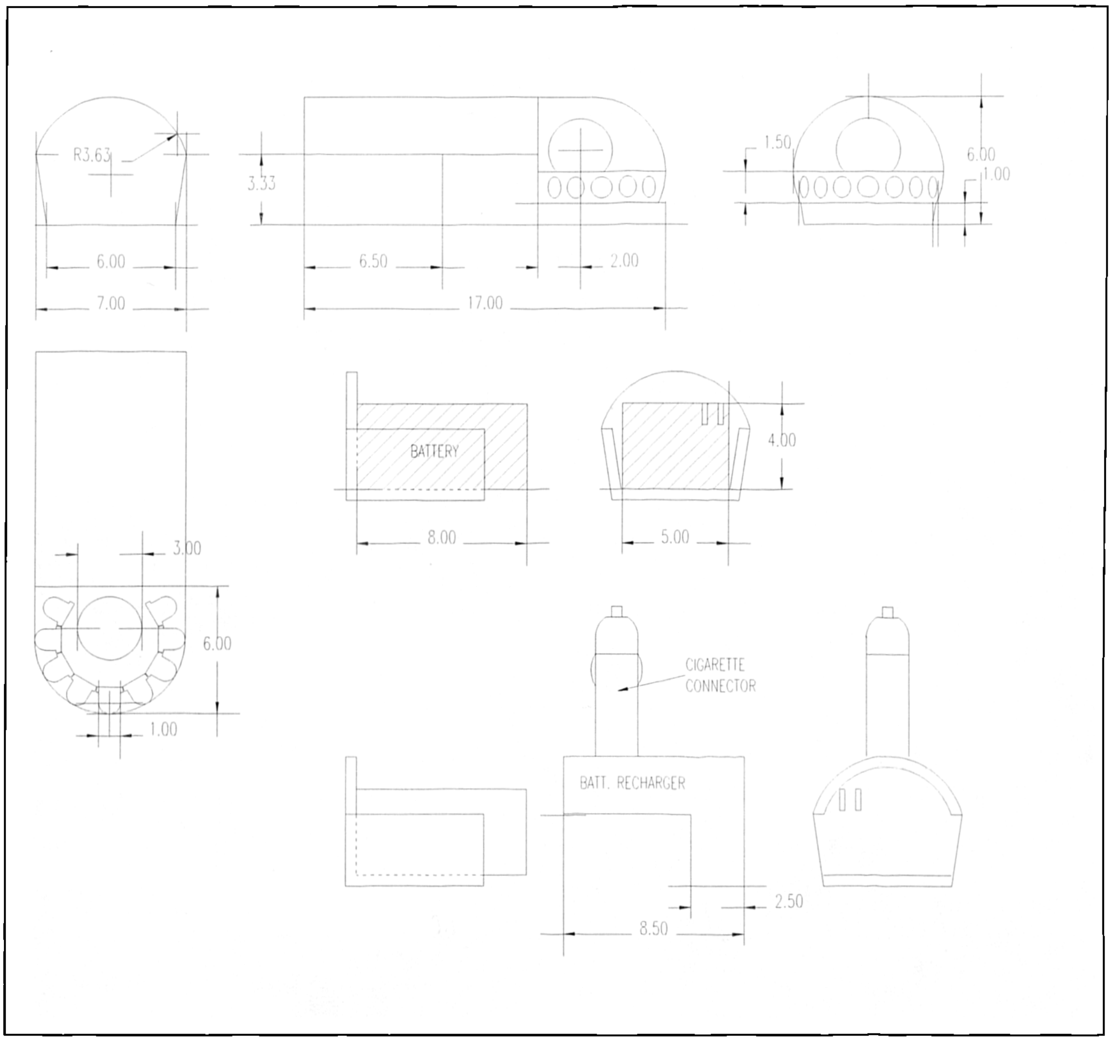
Charger gets it's power via a connector to the car's cigarette lighter.

This car torch is easy to use and a pleasure to hold in your hand.

A.2.2 Presentation Drawings

Top left : Torch rendering
Bottom : Dimension drawing





A.3 Assessment of the design outcome

A.3.1 Assessment sheet

Assessment Criteria	Lowest Mark						Highest Mark
Aesthetics	[]	[]	[]	[]	[]	[]	[]
Consideration of user interface	[]	[]	[]	[]	[]	[]	[]
Originality of idea	[]	[]	[]	[]	[]	[]	[]
The way that the final proposals were presented	[]	[]	[]	[]	[]	[]	[]

A.3.2 Assessment Data

The first team (with face-to-face meetings)

Criteria	Examiner 1	Examiner 2	Examiner 3	Examiner 4	Examiner 5	Average	δ
Aesthetics	6	6	6	5	5	5.60	0.55
Consideration of User interface	5	5	6	3	4	4.60	1.14
Originality of idea	6	4	6	5	4	5.00	1.00
Presentation	4	4	4	3	3	3.60	0.55

The second team (with electronic meetings)

Criteria	Examiner 1	Examiner 2	Examiner 3	Examiner 4	Examiner 5	Average	δ
Aesthetics	4	2	2	3	3	2.80	0.84
Consideration of User interface	4	2	3	2	3	2.80	0.84
Originality of idea	4	2	3	2	4	3.00	1.00
Presentation	4	2	5	3	3	3.40	1.14

Appendix B: Results of the Usability Experiment

B.1 Examples of transcribed dialogues

The dialogues are represented using the following table.

Unit P1S	Transcribed speech	Task Type							Trn	Time (sec)	SS W
		1	2	3	4	5	6	7			

Each row represents a dialogue segment. Each column shows the following information.

- Unit(ID): This column indicates pair ID and condition used. (e.g. P1S indicates Pair 1 using Shared Stage Mode. P8F indicates Pair 8 using File exchange Mode.)
- Transcribed speech : Actual dialogue between participants.
- Task type: The number indicates a interaction category described in the chapter 6 section 6.4.2.4. The V mark indicates the occurrence of the interaction category in the dialogue segment.
  1. Planning process
  2. Model-centred interactions
  3. Method-centred interactions
  4. Co-ordination of process
  5. Interactions related to system functions
  6. Interactions related to shared outcome
  7. Digression and others
- Turn: The number of turns in a dialogue segment
- Time: Time taken for a dialogue segment
- SSW: The S mark indicates that the shared stage window was used in the dialogue segment.

The frequency of each measure was counted and summed at the end of the table. The following table shows an example transcription.

Unit P1S	Transcribed speech	Task Type							Trn	Time (sec)	SS W
		1	2	3	4	5	6	7			
01	00:00 N: So what are you going to do? B: Well er... N: Ben? B: Well Nadia I think I will do the uh.. the chair N: The chair and I'll do the table. B: If I finish the chair quickly, I can give you a hand. N: OK, you can give me a hand making drawers or something. just copy them, OK I will do the table. 00:30	V							6	30	
02	01:08 B: I think my chair is a bit too small for your table so I'm going to enlarge my chair slightly OK. N: OK B: It looks to be about sort of a fifth of the length of the table. That's about a fifth I think. Two squares of your table so ten squares long, so that's about right. 01:30						V		2	22	
ML	01:59 B: Ah it's not symmetrical... I put the wrong button, put the wrong function. That's the one I want.					V			0	5	
ML	02:24 N: How do I get this off?					V			0	3	
03	03:10 B: My chair leg's two squares high. OK N: Two squares high B: Yeah so just to give you an idea for your lengths of your table N: Just two squares? B: It looks like the legs are slightly shorter than the er actual proportions of the er... two squares is a bit long maybe in the drawing there. N: This now looks .. B: I think the legs are actually as long as the seat is wide from the orthographic views, drawings. 03:45		V						6	35	
04	04:01 N: Why is your leg in the air? B: My leg is too high actually. I'll move it down a bit. B: I'm just gonna reference some of your bits so that I can see them with my other things. N: As long as you reference them and don't take them away. B: I can see where you are now. 04:35				V		V		2	34	S
05	05:39 B: Actually it's quite good not having your table in my viewa at the moment, 'cos otherwise I wouldn't be able to see what I was doing with my stack things. 05:45					V			0	6	S
06	07:06 B: How are you making solid? N: I'm making the drawers, yeah. B: Oh right. 07:13				V				2	7	
07	07:57 B: I finished my chair. Can you see my chair? Have a swivel round. Have a look at it. N: Turn it round. Hmm, why is it in the end of the table. B: It is .. N: It's in my table. B: I will group together and put it in position there. So you've got an idea of where it is so.. N: Well OK.			V	V		V		12	63	S

	B: But you have to reference your table across to have a look N: Have you referenced everything of mine across? B: No. I haven't referenced anything yet. N: No B: I have referenced one thing - a leg. Can you reference the table top? should reference my things. Do you know which number your table top is? N: No I don't. B: Go on then. I think it's number 6. Or 0 maybe. Yeah. I'll move my chair down. 09:00										
ML	10:00 B: Just gonna put back my chair straight or it will be out of line.			V					0	5	
ML	10:39 N: It really looks funny.			V					0	3	
08	10:59 B: Shall I make a little knob for your... N: I've done it. B: Have you? N: I know I'm boring aren't I? B: Make the chair a bit lower... B: Thers's not much room between your draws for the chair. N: Thin person. B: ...those draws. N: Very very thin person. It doesn't matter. Very very thin. 11:48	V	V	V	V				6	49	
09	11:58 B: I knicked something of yours then. N: Give back. Ah, you know how I could have done this? Doing first the set and then doing the cylinders. Now I just have to do the cylinders one by one. B: First. How do you mean? Oh you... N: Yeah. 12:24			V			V	V	3	26	
10	13:19 N: Your chair looks a bit low. B: Yeah, it is a bit. N: A thin small person with a giant's desk. B: Are you still making bits for your desk? N: No, I've finished that. B: I thought I'd referenced everything but I can only see one of the knobs on the... N: One knob only? B: On the right hand bank of drawers. I can only see one knob on my front elevation. I thought I'd referenced everything. N: Ah B: Two missing knobs. Is one of your knobs still selected? N: Yeah B: Um well. N: You mean just unpick them or pick them? B: Maybe we ought to make your drawers a bit narrower. If you group them and sort of distort them. N: I did group them actually. B: Just squash them up a little bit in the X axis. I'll do a bit N: Ok. Let me do it. B: No I'll do a bit. No I haven't got it. N: Well then you have to take off my stuff. B: Let me do it. I'll tell you when it's in the right place. N: OK 15:08			V	V		V		20	109	S
11	15:24 N: Ooo, this is hard. Can't do it. B: No?. N: No. Oh gosh. No can't do it. B: Which view are you doing it in? In the... N: The front view. Can you see what's happening?			V		V	V		22	209	S

	B: Yeah, There going all over the place. N: I don,t know what's wrong with them. Let me try doing it this way view - on the top bit. <TJ: Reset with 1,1,1.> N. Oh. OK here? <TJ: Yes.> N: Can I do it again? <TJ· Yes, using...with the mouse.> N: A middle mouse? <TJ: You want to...the front view... > B: ...the X axis... N: Ooo, same. <TJ: You've got to use repetition.> N: What must I use? Ah it's hard work! B: Have we got the other set on the other side...do you know what numbers they are? N: No B: Is there any way of finding out what numbers the other draws are? <TJ: Press 0 and you will have...> B: Which one? <TJ: Press object 0.> B: Why's that? <TJ: Then the 3.> B: Then the 3. Oh I see. Which button? <TJ: Any one.> B: Oh I see, right. Then it lights up red. Oh I see. That's all the knobs there...and that's the drawer there. N: I can't... B: Right I'm going to nick all those drawers ok? N: ok. Wait just a second. Let me press 1,1,1 again. Don't move. Wait, wait, wait. B: You need to get the set of drawers. I'm gonna knick the other set of knobs as well. N: Ok, there you go. You can nick... B: No I'm not going to touch the ones on the left. I'm only gonna touch the ones on the right. N:OK B: I'l get the knobs on the right. N: I think maybe we should have grouped the knobs as group B: I've taken one of the other knobs by mistake. N: Give it back. B: I think it's 42. I'll give it back. N: OK 18:53										
12	19:01 B: The other one must be...36. That's it. Ok, I've got all your knobs and all your drawers on that side. I'll do the big view. Get rid of this thing. Right here goes. B: We've got the of the chair. Just slowing down this thing. N: What are you doing, you making my knob very very big. 19:50				V				0	49	S
13	20:11 N: You know I find it...did you...Oh no...did you do 1,1,1? B: I did. N: That's not how it looks. Very very big knobs. B: I thing maybe I should group them before I do this. N: What did you try to do? B: I tried to change their width. N: Why? They were fine like they were. B: I wanted them to be wider... narrower. Oh it's going mad again. <TJ: Are you having some sort of technical problems?> B: No <TJ: It should be more like this.> <TJ: Do you remember the scale you did on the knob?					V	V		7	72	

	N: No 21:23										
14	22:04 B: That is a much more sensible size. N: So Ben what do you want to do? Explain to me please. B: I just wanna try to make those drawers a little bit narrower, so there's a bit more space. N: Try what I did first. B: What did you? N: Well I tried to use a non-proportional scale. B: That's what I tried. N: It's funny isn't it. It just slips out. B: I thought I was doing non-proportional, and then I thought I was doing the middle... <TJ: I think it's because of your pivot point.> B: Oh I see. N: I don't. B: I think if you do it away from the origin point... <TJ: Draw...> B: Then pick them. <TJ: Without..> B: Without the knobs. N: Hmm B: Change scale. Now the origin point of the draws is in the middle of the draws isn't it? Maybe it's there. N: Do you see what happens...if you try and do... B: Oh it's working alright now. <TJ: Change the position.> B: That's nice. I think that's about the right size. N: How do you do it then? B: Nadia change the position. Go away. Your origin point... N: What's wrong with my origin point? B: Well if your moving, if your zooming, if your changing the side a long way away from...your origin point you've got tiny movements of the cursor make a huge difference to the... N: Yeah... B: Make a huge difference to the site of the object. N: Take my... what I will do is... you copy that set of drawers which you just did. I'll delete mine and then you just pass it over to me. B: ok. N: Or something. Shall I delete mine then? B: OK N: Pass the set... B: Um OK. N: You haven't taken away the knobs? The knobs are same. B: Knobs are same yea. One of your knobs has disappeared. 24:56			V		V	V		28	172	
15	25:04 N: Have you set any of the drawing... the thing yet? B: No N: You haven't copied it yet? B: I have copied and pasted it but I don't know how to send it to you. N: just uh.. do I have to take it from you? B: I think so yeah. N: Oh B: I don't know what number it is. If you...click on the drawers... N: Which button...oh...um...is it the second button yeah? To take away. Teleport, that one it's not working. I'm gonna activate my screen. B: I'll put them there. N: So what number is that? B: I don't know. It should light up on the tree but I can't see what the tree... N: no.. is it this one? I am taking something from you. What did I take? B: You took the drawers.			V	V	V	V		20	71	S

	N: Oh I took drawers that's right I took them. B: Where have you put them? N: yeah I want them back. B: Where have you put the drawers though? N: I took them to my... B: I just put a set of drawers on your side. N: No no it's crashed. <TJ: OK that's enough.> 26:15										
	Total 15 Dialogue Segment --- Frequency Total	2	2	9	7	7	9	1	136	970 sec	7

Unit P1NS	Transcribed speech	Task Type							N of Turn	Time (sec)	Using SSW
		1	2	3	4	5	6	7			
01	33:16 N: I'm doing the bottom bit. I'll do the top bits. You can do the nitty gritty ones. B: What the arial one or the legs? N: You can do that. Which is quite particular and then if you want, I can do the little cylinders. B: Work out what the dimensions are so we've got no problem getting it the right size anyway.Maybe, you can take it across and orientate it to the right position to go onto your bit. N: What the ariel bit? Yeah that's fine OK. 33:55	V		V					4	39	
ML	35:17 B: I've forgotten how to rotate it. 36:02 <TJ:You've got to reset it to 000.> 36:11 B: I think I was thinking you've got to put in both. 37:17 B: It's gone back to... I didn't want it... 38:23 N: That's alright. Definitely. 38:32 <TJ: You know proportion scale. Not quite 5. 3 or point 5. I think that's what it was.>					V			0	0	
02	42.20 N: How are you getting on then? B: Oh alright. It,s a bit slow. You see those little things inside the aerial - I think I approximatded them to being 0.2 of a metre cubes rather than being... 'cos they really should fit round the aerial. I'm not sure really. I just give them a bit of clearance there rather than taking them right up. N: I don't think you have to be really exact. B: Also I've got to ofset the bottom block a bit. N:Well I've nearly finished mine. I just have to do the two cylinders and then that's it. B: Well all I've done is the aerial. I tell you what I'll save mine and then you can grab it, stick it on your bit. N: Yeah OK B: I don,t understand why we've got two sets of lines on the plan. I think I'll group mine together first before I er... turn it over. 43:44			V	V				7	144	
03	44:03 N: That really looks strange. B: What does? N: My cylinder. It really really looks strange. B: I'll call it arial. <TJ: You have to add T3D as the extension.>					V	V		4	107	

	B: Oh do you? Is it T sensitive? <TJ: Yeah. No no, just aerial.> B: It's 1 aerial. 2 t3d if you want to look at it. N: Yeah, I will in a second. I'm just getting my cylinders sorted first. They don't seem to want to get sorted. I don't know why. <TJ: You have to use reciprocation sorry. Again there is the...> N: What is it...1 1 1 ? <TJ: No you have to rotate 000.> N: Oh 000 <TJ: Also you have to reset 1 1 1... and scale also 1 1 1.> N: I see now. I want to rotate round... 090 0... I'm not sure if this is going to work...It's not working. N: Not working....I'll try this one...Yes, got it. 45:50										
04	46:13 N: No I don't want it like that. B: If you save yours OK. I can have a look at that. N: Are you like finished? I'm gonna save it as....save it as what? B: body.t3d N: Saved it. You probably don't want that little thing that I saved. Why does it do that? 46:43			V			V		4	30	
05	46:53 N: I like you like that but I want you thinner. B: I got yours, but I,m not...lets have a look... N: Oh no .. yes got you. B: I've managed to get them both together now with the append. I kept on doing open, but I hadn't done append. I've got to rotate my aerial because it's flat. B: I'll group it...It's not accurate. 48:20					V	V		4	87	
06	48:38 N: Have you don't that interesting part? B: I've done the arial. N: Have you done put on the line. B: I haven't yet no. I've just turned it round the right way. I'm gonna try and put it on there. B: It's almost in the right place. I,ve just got to move it. 48:50			V	V				4	12	
07	48:58 B: If you do a.. N: I am gonna get your arial in. B: If you go and append on my files to yours you'll keep yours and get my arial. If you just use open you'll lose yours. N: I got your things. B: What's that bit floating in the ... below the whole thing? N: Below? B: There's a funny sort of diamond shape thing floating. N: Where? On my stuff? B: Yeah N: A diamond shape floating - where? B: Look on perspective view. It's all on the same size the.... on any of the views. I didn't draw that. N: Where? What is this diamond shape thing? B: It's like a sort of... <TJ: You cant see each others window....and its not the same.> B: Well I just... when I imported yours it was there. When I appended yours it was there, you must have rubbed it out. N: Yeah, yeah. B: Shall we do some feet? N: I just gonna rotate this.....rotate it um.....0,0,0. B: Well thanks for sending me your body. N: That doesn't sound nice. Yes, got it, let's go. B: Maybe rub that thing out, I don't know what it is.			V			V		20	72	

	N: I think I'm a genius. Very genius. B: Actually I'm not going rub that out I'm gonna send it to you. I,m gonna save it as a new name though - that thing. 50:10										
08	50:15 B: How do you save just the highlighted thing? Save active. N: There you go. B: Save active. I'm gonna call it lostbit. B: If you open up lostbit.t3d, you will find that thing I was talking about - lostbit. N: Oh I see. You know why? B: Why? N: I'll explain it to you. I was doing a cylinder. And you said save it. I saved it at that moment and you got that funny little thing that was floating in the air. B: So have you find it? Have you opened up lostbit.t3d? N: I am gonna send you what I've done now. And you'll see one piece of art work. All right? B: But have you seen the bit I sent you? N: Let me save my bit ok. Save - I'm gonna say - complete body. B: Can we have long name like that? N: I'll just say combody.t3d. ok. B: You haven't moved it to a different place, so I can open that and it will be the new body in the same place. N: It will be the - Yeah - it will be perfect. I gonna append - do I have to append your one then to see what you have done? B: If you do an append lostbit, lostbit should appear on your window. N: lostbit - OK. Open . Let's see what happens. B: I'm gonna append combody. N: You sent me back the thing. Is that what you did? I am gonna cancel it. B: I got your combody now. I like your feet. 51:26			V			V		18	71	
	Total 8 Dialogue segment	1	0	6	2	3	5	0	65	562	

B.2 Questionnaire

The following questions ask you to consider how the system supported a collaborative 3D modelling process. Some questions are statements that you may agree or disagree with, other questions have positive and negative slants. Please tick the appropriate boxes below or write a short response.

Experience of multi-user systems

- Have you ever used any of the following real-time multi-user systems?

	Yes	No
- Networked Game	[ ]	[ ]
- Video Conferencing	[ ]	[ ]
- Shared white board system	[ ]	[ ]
- Other real-time collaborative systems, please write details if any ______________		

3D Modelling software

- What kinds of 3D modelling software have you used?

____

- What is your favourite 3D modelling software among them?

____

- How long have you used that software? __________ Month

About the first condition: [] NS [] S

	Positive		Neutral		Negative
- I am satisfied with what we produced.	[]	[]	[]	[]	[]
- I am satisfied with the collaboration process.	[]	[]	[]	[]	[]
- We were able to collaborate effectively.	[]	[]	[]	[]	[]
- I knew what my partner was working on during the collaborative session.	[]	[]	[]	[]	[]
- This multi-user system is different from conventional single user 3D modelling system.	[]	[]	[]	[]	[]
- Other comments ______________________					

About the second condition: [] NS [] S

	Positive		Neutral		Negative
- I am satisfied with what we produced.	[]	[]	[]	[]	[]
- I am satisfied with the collaboration process.	[]	[]	[]	[]	[]
- We were able to collaborate effectively.	[]	[]	[]	[]	[]
- I knew what my partner was working on during the collaborative session.	[]	[]	[]	[]	[]
- This multi-user system is different from conventional single user 3D modelling system.	[]	[]	[]	[]	[]
- Other comments ______________________					

User Interface of the system

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
- The system speed interrupted the 3D modelling process.	[]	[]	[]	[]	[]
- Navigation of 3D modelling views (dolly, track, tumble) was easy to use.	[]	[]	[]	[]	[]
- Synchronised camera view was easy to use	[]	[]	[]	[]	[]
- Synchronised camera view was valuable to perform the task.	[]	[]	[]	[]	[]
- Locking mechanism for the control of shared stage view made a conflict in my workspace.	[]	[]	[]	[]	[]
- Telepointer palyed an important role in communicating with my partner.	[]	[]	[]	[]	[]
- Red outline indicating who has the control provided useful information about the state of shared workspace.	[]	[]	[]	[]	[]
- Referencing and transporting by object nodes was a valuable function for collaborative modelling.	[]	[]	[]	[]	[]
- The user interface of referencing by object nodes was easy to understand and use.	[]	[]	[]	[]	[]
- The little rectangle under object nodes indicating reference state provided valuable information about the state of shared workspace.	[]	[]	[]	[]	[]
- The size of Shared Stage Window was too small.	[]	[]	[]	[]	[]
- Bigger Shared Stage Window would be better despite hiding a part of modelling views.	[]	[]	[]	[]	[]
- The Shared Stage Window improved the quality of output.	[]	[]	[]	[]	[]
- The user interface of the Shared Stage Window was difficult.	[]	[]	[]	[]	[]

- Other comments about the user interface ______________________

Efficiency

	Positive		Neutral		Negative
- The Shared Stage Window allowed us to perform parallel tasking.	[]	[]	[]	[]	[]
- The Shared Stage Window allowed us to allocate the work effectively.	[]	[]	[]	[]	[]
- The Shared Stage Window facilitated efficient exchange of object data	[]	[]	[]	[]	[]
- The Shared Stage Window allowed us to integrate individual output effectively.	[]	[]	[]	[]	[]

- Other comments about efficiency ______________________

Preference

Condition NS : Using file exchange for collaboration
Condition S : Using the Shared Stage Window for collaboration

	Very Easy	Easy	Neutral	Difficult	Very Difficult
- How do you find the collaboration process in condition F?	[]	[]	[]	[]	[]
- How do you find the collaboration process in condition S?	[]	[]	[]	[]	[]

	Condition NS	Condition S
- Which condition better supported collaborative modelling tasks?	[]	[]
- Which condition was easier to use for group work?	[]	[]
- Which condition do you prefer?	[]	[]

- Other comments about preference ______________________________

Thank you for your participation.

Appendix C: Source Code of the Syco3D System

This appendix shows the structure of the Tcl source code of the Syco3D system. First, a list of files and procedures within each file are presented. Scripts and procedures are presented in groups to illustrate related functions. Some source codes for the procedures of the list are presented as an example.

C.1 List of files and procedures

Syco3D.tcl

This file contains start-up scripts that open application window and load other files.

Menu.tcl

This file manages the construction of menu structure and interface. The procedures included are as follows.

```
proc mMenuSetup { menubar }
proc mMenu { label }
proc mPaletteMenu { label }
proc mMenuCommand { menuName label command }
proc mMenuCheck {menuName label var { command {} } }
proc mMenuRadio {menuName label var { val {} } { command {} } }
proc mMenuSeparator {menuName}
proc mMenuCascade {menuName label}
proc MAddPaletteIcon {palette_icon_row_widget m n command}
proc MCreateIconRow {palette_icon_widget n}
proc MCreateIconWidget {menu_widget}
```

TkInteractor.tcl

This file is a generic interactor of the Visualisation Toolkit and contains procedures that manage interactive change of 3D view.

```
proc BindTkRenderWidget {widget}
proc GkBindTkRenderWidgetOriginal {widget}
proc Render {}
proc UpdateRenderer {widget x y}
proc Enter {widget x y}
proc StartMotion {widget x y}
proc EndMotion {widget x y}
proc Rotate {widget x y}
proc Pan {widget x y}
proc Zoom {widget x y}
proc Reset {widget x y}
proc Wireframe {}
proc Surface {}
proc Pick {widget x y}
```

vtkInt.tcl

This file is a generic interactor of the Visualisation Toolkit and called in TkInteractor.tcl.

```
proc vtkInteract {}
```

lock.Tcl

This file contains procedures that manage lock states of the Synchronised Stage View.

```
proc lock_request {}
proc lock_callback {state}
proc lock_release {}
proc set_delay {}
proc reset_delay {}
proc GkBindTkRenderWidget {widget}
```

colortable.tcl

This file contains scripts that create colour table.

init.tcl

This file contains scripts and procedures that handles initialising default properties and user ID.

```
proc MAssignColorToProperty {p color_value}
```

grid.tcl

This file contains procedures that manages grid reference.

```
proc MResetGridValues {}
proc MCreateGridPlane {}
```

layouts.tcl

This file contains procedures that handles creating and resizing windows for 3D view.

```
proc MInitializeWindowProperty { }
proc MCreateNewWindow { name }
proc MToggleWindowSize { name }
proc MDeleteViewWindow { name }
proc MResetBindTkRenderWidget {}
proc MDisplayCoordinate {w x y}
proc MDisplayRefInfo {info}
```

object.tcl

This file contains procedures that manage creating primitive objects.

```
proc MCreateCone {}
proc MDoCreateCone {xyz id winname}
proc MDisplayPrompt {prompt_text}
proc MCreateCube {}
proc MDoCreateCube {xyz id}
proc MCreateSphere {}
proc MDoCreateSphere {xyz id}
proc MCreateCylinder {}
proc MDoCreateCylinder {xyz id winname}
```

profile.tcl

This file contains procedures that handle conversion of coordinates and addition of objects into render windows.

```
proc MDisplayToWorld {widget x y}
proc MWorldToDisplay {x y z}
proc MAddActorToAllRenderWindow {actor id}
proc MRemoveActorFromAllRenderWindow {actor id}
proc MRemoveActorFromCurrentRenderWindow {actor}
proc MAddActorToCurrentRenderWindow {actor}
proc MUpdateAllRenderWindow {id}
proc MUpdateMySharedRenderWindow {}
proc MResetAllSharedRenderWindow {}
proc MSyncAllSharedRenderWindow {}
```

pick.tcl

This file contains scripts and procedures that manage object selection.

```
proc MStartPickObject {}
```

```
proc MRemovePickRubber {id}
proc MCreatePickRubber {id}
proc MPickActor {widget x y}
proc MPickActorByPixel {x y usrid}
proc MPickActorByBoundingBox {ox oy x y usrid}
proc MDrawPickRubber {widget x y}
proc MListActiveActors {id}
proc MListAllActors {usrid}
proc MStartPickNothing {}
proc MPickNothing {usrid}
proc MStartPickAll {}
proc MPickAll {usrid}
proc MStartPickLast {}
proc MPickLast {usrid}
proc MAssignLastActors {actors usrid}
```

delete.tcl

This file contains procedures that manage deleting objects.

```
proc MStartDeleteActiveActors {}
proc MDeleteActiveActors {id}
proc MStartDeleteCopiedActors {}
proc MDeleteCopiedActors {id}
proc MDeleteActor {actor active_collection usrid}
proc MStartDeleteAllActors {}
proc MDeleteAllActors {id}
```

xform.tcl

This file contains procedures that manage transformation of obejcts, such as move, rotate, scale, non proportional scale and change origin.

```
proc MStartMove {}
proc MMove {lxyz xyz b id}
proc MStartScale {}
proc MScale {lxyz xyz b id}
proc MStartNonpScale {}
proc MNonpScale {lxyz xyz b usrid}
proc MStartRotate {}
proc MRotate {lxyz xyz b id}
proc MStartSetOrigin {}
proc MSetOrigin {xyz id}
proc MSetActorOrigin {actor xyz}
```

edit.tcl

This file contains procedures that manages copying, pasting and grouping.

```
proc MStartCopy {}
proc MCopy {usrid}
proc MCopyActor { actor collection }
proc MInitializeCopiedActors {usrid}
proc MStartPaste {}
proc MPaste {usrid}
proc MPasteActor {actor collection collection_s usrid}
proc MStartGroup {}
proc MGroup {usrid}
proc MStartUnGroup {}
proc MUnGroup {id}
proc MMultiplyActorsTransformation {actor partactor}
```

objectdisplay.tcl

This file contains procedures that manage representation of objects.

```
proc MRepresentationToWire {}
proc MRepresentationToSurface {}
proc MChangeActorRepresentation {actor representation}
proc MRepresentationToPoints {}
```

sharedstage.tcl

This file contains procedures that manage creation of the Shared Stage Window.

```
proc MSharedStage {}
proc MWithdrawSharedStage {}
```

hrepresent.tcl

This file contains procedures that manage information and interface components in the Data Structure Diagram

```
proc MScrolledCanvas { c width height region }
proc MResetParticipants {}
proc MDeleteAllParticipants {}
proc MDrawParticipant { pnum pcol x y }
proc MConferenceEventHandlerOn {}
proc MUpdateStructureTree {}
proc MUpdateStructure {id color xposition yposition}
proc MPrintStructure {actor depth dist orig color level}
proc MCountLinesOfStructure {actor id}
proc MAddActorToAlmostAllRenderWindow {actor usrid}
proc MRemoveActorFromAlmostAllRenderWindow {actor usrid}
proc MChangeActorsOwnership {actor usrid}
proc MRecursiveOwnerInfoChange {actor usrid}
proc MStartRubberLineOnCanvas {w x y actor}
proc MDrawRubberLineOnCanvas {w x y}
proc MChangeOwner {w x y actor}
proc MStartLinkActor {w x y actor color}
proc MLinkActor {actor usrid color}
proc MUnLinkActor {actor}
proc ldelete { list value }
```

file.tcl

This file contains procedures that handles data save and retrieval. proc MStartFileSaveActive {}

```
proc MFileSaveiActive { usrid }
proc MStartFileSave {}
proc MFileSave { usrid }
proc MFileSaveHeader {fileId}
proc MFileSaveActor {actor collection fileId}
proc MStartFileOpen {}
proc MFileOpen { usrid }
proc MReadSegment { fileId collection usrid}
proc MAddSegment {fileId collection usrid}
proc MStartTempFileSave {}
proc MTempFileSave { usrid }
proc MStartTempFileOpen {}
proc MTempFileOpen { usrid }
proc MStartFileAppend {}
proc MFileAppend { usrid }
proc MDisplayPoint {} {
proc MRelocateOrigin {org act}
```

gkenv.tcl

This file contains scripts that manages synchronisation of the Shared Stage View using the GroupKit's environment feature.

```
proc MUpdateSharedCamera {}
```

startup.tcl

This file contains a startup script that creates default modelling views, the Shared Stage window and multi-user cursors.

C.2 Example source code

C.2.1 Syco3D.tcl

```
# This file contains start-up scripts that open application window and load
# other files.

wm withdraw .
toplevel .top -width 950 -height 950
wm title .top "ider3D for collaborative modelling"
pack propagate .top false
wm minsize .top 300 200
wm maxsize .top 950 950
wm geometry .top 950x950+125+35

toplevel .palette -width 50 -height 950
wm title .palette "Palette"
pack propagate .palette false
wm minsize .palette 50 50
wm maxsize .palette 50 950
wm geometry .palette 50x950+10+35

# env(I3D) is set to a root directory of the Syco3D system
source $env(I3D)/multi/menu.tcl
source $env(I3D)/multi/TkInteractor.tcl
source $env(I3D)/multi/lock.tcl

set frame_icons [frame .top.icons -borderwidth 1 -width 800 -height 20 \
      -relief sunken]
pack .top.icons -side top -fill x -expand true -anchor nw
set frame_promptlabel [label .top.icons.label -font $Menu(font)]
pack $frame_promptlabel -side left
set frame_promptentry [entry .top.icons.entry -font $Menu(font)]
pack $frame_promptentry -side left

# frame for layer information
set frame_layers [frame .top.layers -borderwidth 1 -width 800 -height 20]
pack $frame_layers -side top -fill x -expand true -anchor nw
message .top.layers.coord -justify right -text "x y z" -font $Menu(font) -
width 800
pack .top.layers.coord -side right

# frame for 3D viewports
set frame_view [frame .top.views -width 1000 -height 900 -borderwidth 1 \
      -relief sunken]
pack $frame_view -expand true -fill both -anchor nw

# create default vtk objects (e.g. property objects etc.)
source $env(I3D)/multi/colortable.tcl
source $env(I3D)/multi/init.tcl

# Create Grid Primitive and initialize
source $env(I3D)/multi/grid.tcl
MCreateGridPlane

# Create View Windows and launch grid primitive
source $env(I3D)/multi/layouts.tcl

# Load other procedures
source $env(I3D)/multi/object.tcl
source $env(I3D)/multi/pick.tcl
source $env(I3D)/multi/delete.tcl
source $env(I3D)/multi/xform.tcl
```

```
source $env(I3D)/multi/edit.tcl
source $env(I3D)/multi/objectdisplay.tcl
source $env(I3D)/multi/sharedstage.tcl
source $env(I3D)/multi/hrepresent.tcl
source $env(I3D)/multi/file.tcl
source $env(I3D)/multi/origin.tcl
source $env(I3D)/multi/gkenv.tcl
source $env(I3D)/multi/startup.tcl
```

C.2.2 lock.Tcl

```
# This file contains procedures that manage lock states of the Synchronised
# Stage View.

proc lock_request {} {
    locks request myLock "lock_callback"
}

proc lock_callback {state} {
    if {$state=="Succeeded"} {
       gk_toAll .sharedstage.f2.canvas itemconfig user[locks owner myLock] \
       -width 3 -outline red
    } else {
    }
}

proc lock_release {} {
    gk_toAll .sharedstage.f2.canvas itemconfig user[locks owner myLock] \
       -width 1 -outline Black
    locks release myLock
}

proc set_delay {} {
    locks delay 5000
}

proc reset_delay {} {
    locks delay ""
}

set motion_flag 0

proc GkBindTkRenderWidget {widget} {
    global motion_flag
    bind $widget <Any-ButtonPress> {
        lock_request
    }
    bind $widget <Any-ButtonRelease> {
        if {[locks owner myLock] == [users local.usernum]} {
           EndMotion %W %x %y
           lock_release
           set motion_flag 0
        }
    }
    bind $widget <Shift-Alt-B1-Motion> {
        if {[locks owner myLock] == [users local.usernum]} {
            if { $motion_flag == 0 } {
                StartMotion %W %x %y
                set motion_flag 1
            }
            Rotate %W %x %y
           MUpdateSharedCamera
           GkEnv_flag set cflag 1
        }
    }
```

```
    bind $widget <Shift-Alt-B2-Motion> {
        if {[locks owner myLock] == [users local.usernum]} {
            if { $motion_flag == 0 } {
                StartMotion %W %x %y
                set motion_flag 1
            }
            Pan %W %x %y
           MUpdateSharedCamera
           GkEnv_flag set cflag 1
        }
    }
    bind $widget <Shift-Alt-B3-Motion> {
        if {[locks owner myLock] == [users local.usernum]} {
            if { $motion_flag == 0 } {
                StartMotion %W %x %y
                set motion_flag 1
            }
            Zoom %W %x %y
           MUpdateSharedCamera
           GkEnv_flag set cflag 1
       }
    }
    bind $widget <KeyPress-r> {
           Reset %W %x %y
           MUpdateSharedCamera
           GkEnv_flag set cflag 1
    }
    bind $widget <KeyPress-u> {
           wm deiconify .vtkInteract
    }
    bind $widget <KeyPress-w> {
           Wireframe
    }
    bind $widget <KeyPress-s> {
           Surface
    }
    bind $widget <Enter> {
           Enter %W %x %y
    }
    bind $widget <Leave> {
           focus $oldFocus
    }
}
```

C.2.3 object.tcl

```
# This file contains procedures that manage creating primitive objects.

set n 0

# Create Cone
proc MCreateCone {} {
    global frame_promptentry myid icon

    MResetBindTkRenderWidget
    $icon(frame,Cone) config -bg red
    MDisplayPrompt "Enter cone location (x y z) : "

    set tmp_value 0
    $frame_promptentry configure -state normal -textvariable tmp_value

    bind vtkTkRenderWidget <ButtonRelease-1> {
        set xyz [MDisplayToWorld %W %x %y]
       # window name looks .top.views.1157_top.window.r1
       regexp {[^_]*_([^.]*)} %W match winname
```

```
        gk_serialize MDoCreateCone $xyz $myid $winname
         MCreateCone
      }
      bind $frame_promptentry <Return> {
        if [regexp {([^,]*),([^,]*),([^,]*)} $tmp_value match x y z] {
            set xyz "$x $y $z"
        } else {
            set xyz $tmp_value
        }
        gk_serialize MDoCreateCone $xyz $myid front
         MCreateCone
      }
      bind all <Escape> {
        MDisplayPrompt ""
        MResetBindTkRenderWidget
        bind $frame_promptentry <Return> {}
        $frame_promptentry delete 0 end
        return
      }
}


# Create Cone Operation run with gk_toAll
proc MDoCreateCone {xyz id winname} {
    global renWin ren n pref
    global actors

    set x [lindex $xyz 0]
    set y [lindex $xyz 1]
    set z [lindex $xyz 2]
    if [catch {expr $x+1}] {return}
    if [catch {expr $y+1}] {return}
    if [catch {expr $z+1}] {return}

    vtkConeSource object(Cone,source,$n)
       object(Cone,source,$n) SetResolution $pref(cone_res)
    vtkPolyDataMapper object(Cone,mapper,$n)
          object(Cone,mapper,$n) SetInput [object(Cone,source,$n) GetOutput]
          object(Cone,mapper,$n) ImmediateModeRenderingOn
    vtkActor object(Cone,actor,$n)
          object(Cone,actor,$n) SetMapper object(Cone,mapper,$n)
          object(Cone,actor,$n) SetProperty prop($id)
          object(Cone,actor,$n) SetPosition $x $y $z
         object(Cone,actor,$n) PickableOn
         object(Cone,actor,$n) SetId $id
# display object for origin
    vtkActor object(Cone,origin,$n)
         object(Cone,origin,$n) SetMapper origin_mapper
         object(Cone,origin,$n) SetProperty origin_prop
         object(Cone,origin,$n) VisibilityOff
         object(Cone,origin,$n) PickableOff

    allActors($id) AddItem object(Cone,actor,$n)
    allActors(shared) AddItem object(Cone,actor,$n)
    activeActors(shared) RemoveAllItems
    activeActors($id) RemoveAllItems
    MAssignLastActors object(Cone,actor,$n) $id
    MListActiveActors $id
    MAddActorToAllRenderWindow object(Cone,actor,$n) $id
    MUpdateStructureTree
    incr n
}

proc MDisplayPrompt {prompt_text} {
      global frame_promptlabel frame_promptentry
```

```
        $frame_promptlabel configure -text $prompt_text
        $frame_promptentry delete 0 end
}
```

C.2.4 xform.tcl

```
# This file contains procedures that manage transformation of obejcts, such as
# move, rotate, scale, non proportional scale and change origin.
proc MStartMove {} {
    global frame_promptentry LastX LastY myid icon

    MResetBindTkRenderWidget
    $icon(frame,Move) config -bg red
    MDisplayPrompt "Enter position (x y z)  "

    set tmp_value 0
    $frame_promptentry configure -state normal -textvariable tmp_value

    bind vtkTkRenderWidget <ButtonRelease> {
       set lxyz [MDisplayToWorld %widget $LastX $LastY]
        switch -exact -- %b {
         1   { set xyz [MDisplayToWorld %widget %x %y] }
         2   { set xyz [MDisplayToWorld %widget %x $LastY] }
         3   { set xyz [MDisplayToWorld %widget $LastX %y] }
         default {}
        }
       MMove $lxyz $xyz %b $myid
    }
    bind vtkTkRenderWidget <B1-Motion> {
       set lxyz [MDisplayToWorld %widget $LastX $LastY]
       set xyz [MDisplayToWorld %widget %x %y]
       MMove $lxyz $xyz 1 $myid
       set LastX %x; set LastY %y
    }
    bind vtkTkRenderWidget <B2-Motion> {
       set lxyz [MDisplayToWorld %widget $LastX $LastY]
       set xyz [MDisplayToWorld %widget %x $LastY]
       MMove $lxyz $xyz 2 $myid
       set LastX %x
    }
    bind vtkTkRenderWidget <B3-Motion> {
        set lxyz [MDisplayToWorld %widget $LastX $LastY]
       set xyz [MDisplayToWorld %widget $LastX %y]
       MMove $lxyz $xyz 3 $myid
       set LastY %y
    }
    bind $frame_promptentry <Return> {
        set lxyz [MDisplayToWorld %widget $LastX $LastY]
       if [regexp {([^,]*),([^,]*),([^,]*)} $tmp_value match x y z] {
           set xyz "$x $y $z"
       } else {
           set xyz $tmp_value
       }
       MMove $lxyz $xyz 0 $myid
    }
    bind all <Escape> {
       MDisplayPrompt ""
       MResetBindTkRenderWidget
       bind $frame_promptentry <Return> {}
       $frame_promptentry delete 0 end
       return
    }
}
```

```
proc MMove {lxyz xyz b id} {
    activeActors($id) InitTraversal
    set actor [activeActors($id) GetNextItem]
    while { $actor != "" } {
       if [regexp {([^,]*),[^,]+,([^\)]*) )} $actor match front end] {
            set actor_origin ${front},origin,${end})
       }
       if {$b == 0 } {
           gk_toAll $actor SetPosition [lindex $xyz 0] \
                     [lindex $xyz 1] [lindex $xyz 2]
           set org [$actor GetOrigin]
           transform SetMatrix [$actor GetMatrix]
           transform SetPoint [lindex $org 0] [lindex $org 1] [lindex $org 2] 1
           transform PostMultiply
           set org_world [transform GetPoint]
           set dx [expr [lindex $xyz 0] - [lindex $org_world 0]]
           set dy [expr [lindex $xyz 1] - [lindex $org_world 1]]
           set dz [expr [lindex $xyz 2] - [lindex $org_world 2]]
           gk_toAll $actor AddPosition $dx $dy $dz
           gk_toAll MRelocateOrigin $actor_origin $actor
       } else {
           set dx [expr [lindex $xyz 0] - [lindex $lxyz 0]]
            set dy [expr [lindex $xyz 1] - [lindex $lxyz 1]]
           set dz [expr [lindex $xyz 2] - [lindex $lxyz 2]]
           gk_toAll $actor AddPosition $dx $dy $dz
           gk_toAll MRelocateOrigin $actor_origin $actor
       }
        set actor [activeActors($id) GetNextItem]
    }
    gk_toAll MUpdateAllRenderWindow $id
}
```

C.2.5 edit.tcl

```
# This file contains procedures that manages copying, pasting and
grouping.vtkActorCollection copiedActors($myid)
vtkActorCollection copiedActors(shared)

proc MStartCopy {} {
    global myid
    gk_serialize MCopy $myid
}

proc MCopy {usrid} {
    global n

    if [catch {copiedActors($usrid) InitTraversal}] {return}

    MInitializeCopiedActors $usrid

    activeActors($usrid) InitTraversal
    set actor [activeActors($usrid) GetNextItem]
    while { $actor != "" } {
       MCopyActor $actor copiedActors($usrid)
         set actor [activeActors($usrid) GetNextItem]
    }
    MUpdateAllRenderWindow $usrid
}

proc MCopyActor { actor collection } {
```

```
    global n

    if { [$actor GetClassName] != "vtkAssembly" } {
        # find vtkobejcts related to the actor
        # Delete object names are described as object(object,type,id)
        if [regexp {object\(([^,]*),[^,]*,([^\)]*)\)} $actor match objname id]
{
           foreach i [info command object(*,source,$id)] {
                if [catch {[$i GetClassName] object($objname,source,$n)} \
                    result] {
                  puts $result
                  return
              } else {
                  vtkPolyDataMapper object($objname,mapper,$n)
                     object($objname,mapper,$n) SetInput \
                         [object($objname,source,$n) GetOutput]
                     object($objname,mapper,$n) ImmediateModeRenderingOn
                  vtkActor object($objname,actor,$n)
                     object($objname,actor,$n) SetMapper \
                         object($objname,mapper,$n)
                     object($objname,actor,$n) SetProperty \
                         prop([$actor GetId])

                  vtkActor object($objname,origin,$n)
                     object($objname,origin,$n) SetMapper origin_mapper
                     object($objname,origin,$n) SetProperty origin_prop
                     object($objname,origin,$n) VisibilityOff
                     object($objname,origin,$n) PickableOff

                     # copy transformation matrix
                     set pos [$actor GetPosition]
                     set rot [$actor GetOrientation]
                     set scl [$actor GetScale]
                     set org [$actor GetOrigin]
                     object($objname,actor,$n) SetPosition \
                            [lindex $pos 0] [lindex $pos 1] [lindex $pos 2]
                     object($objname,actor,$n) SetOrientation \
                            [lindex $rot 0] [lindex $rot 1] [lindex $rot 2]
                     object($objname,actor,$n) SetScale \
                            [lindex $scl 0] [lindex $scl 1] [lindex $scl 2]
                     object($objname,actor,$n) SetOrigin \
                            [lindex $org 0] [lindex $org 1] [lindex $org 2]
                       object($objname,actor,$n) SetId [$actor GetId]
                     if { [$collection GetClassName] == "vtkAssembly"} {
                         $collection AddPart object($objname,actor,$n)
                     } else {
                         $collection AddItem object($objname,actor,$n)
                     }
                   incr n
              }
           }
        }
      } else {
        set _n $n
        incr n
        vtkAssembly object(Group,assembly,$_n)
        [$actor GetParts] InitTraversal
        set part [[$actor GetParts] GetNextItem]
        while { $part != "" } {
            MCopyActor $part object(Group,assembly,$_n)
              set part [[$actor GetParts] GetNextItem]
        }
        object(Group,assembly,$_n) SetProperty prop([$actor GetId])

        vtkActor object(Group,origin,$_n)
```

```
            object(Group,origin,$_n) SetMapper origin_mapper
            object(Group,origin,$_n) SetProperty origin_prop
            object(Group,origin,$_n) VisibilityOff
            object(Group,origin,$_n) PickableOff

        # copy transformation matrix
        set pos [$actor GetPosition]
        set rot [$actor GetOrientation]
        set scl [$actor GetScale]
        set org [$actor GetOrigin]
        object(Group,assembly,$_n) SetPosition \
               [lindex $pos 0] [lindex $pos 1] [lindex $pos 2]
        object(Group,assembly,$_n) SetOrientation \
               [lindex $rot 0] [lindex $rot 1] [lindex $rot 2]
        object(Group,assembly,$_n) SetScale \
               [lindex $scl 0] [lindex $scl 1] [lindex $scl 2]
        object(Group,assembly,$_n) SetOrigin \
               [lindex $org 0] [lindex $org 1] [lindex $org 2]
        object(Group,assembly,$_n) SetId [$actor GetId]

        if { [$collection GetClassName] == "vtkAssembly"} {
            $collection AddPart object(Group,assembly,$_n)
        } else {    ;# in case of vtkActorCollection
            $collection AddItem object(Group,assembly,$_n)
        }
    }
}
```

C.2.6 sharedstage.tcl

```
# This file contains procedures that manage creation of the Shared Stage
# Window.

toplevel .sharedstage
wm title .sharedstage "Shared Stage Window $myid"

gk_defaultMenu .sharedstage.menubar
pack .sharedstage.menubar -side top -fill x

# Create TkRenderWidget
frame .sharedstage.f1
vtkTkRenderWidget .sharedstage.f1.r1 -width 350 -height 350
    GkBindTkRenderWidget .sharedstage.f1.r1
pack .sharedstage.f1.r1 -side left -padx 3 -pady 3 -fill both -expand t
pack .sharedstage.f1 -fill both -expand t -side left

set renWin(shared,$myid) [.sharedstage.f1.r1 GetRenderWindow]
set ren(shared,$myid) [vtkRenderer ren_sharedstage]
$renWin(shared,$myid) AddRenderer $ren(shared,$myid)

$ren(shared,$myid) SetBackground 0.33 0.39 0.42

$ren(shared,$myid) AddActor planeActor
set cam [$ren(shared,$myid) GetActiveCamera]
$ren(shared,$myid) AddActor axesActor
$renWin(shared,$myid) Render

wm withdraw .sharedstage

proc MSharedStage {} {
    wm deiconify .sharedstage
}
```

```
proc MWithdrawSharedStage {} {
    wm withdraw .sharedstage
}
```

C.2.7 hrepresent.tcl

```
# This file contains procedures that manage information and interface
# components in the Data Structure Diagram
proc MScrolledCanvas { c width height region } {
    frame $c
    canvas $c.canvas -width $width -height $height -scrollregion $region \
       -xscrollcommand [list $c.xscroll set] \
       -yscrollcommand [list $c.yscroll set]  -relief sunken -borderwidth 2
    scrollbar $c.xscroll -orient horizontal \
       -command [list $c.canvas xview]
    scrollbar $c.yscroll -orient vertical \
       -command [list $c.canvas yview]
    pack $c.xscroll -side bottom -fill x
    pack $c.yscroll -side right -fill y
    pack $c.canvas -side left -fill both -expand true
    pack $c -side top -fill both -expand true
    return $c.canvas
}
MScrolledCanvas .sharedstage.f2 350 350 {0 0 1000 5000}

set tpos 20

proc MResetParticipants {} {
    global tpos Color
    MDeleteAllParticipants
    set tpos 20
    MDrawParticipant [users get local.usernum] [users get local.color]\
        $tpos 20
    foreach i [users keys remote] {
       set tpos [expr $tpos + 180 ]
        MDrawParticipant [users get remote.$i.usernum] \
              [users get remote.$i.color] $tpos 20
       vtkActorCollection allActors($i)
       vtkActorCollection copiedActors($i)
       vtkActorCollection activeActors($i)
       vtkPicker picker($i)
       MAssignColorToProperty prop($i) $Color([users get remote.$i.color])
    }
}

proc MDeleteAllParticipants {} {
    .sharedstage.f2.canvas delete -tag Participant
}

proc MDrawParticipant { pnum pcol x y } {
    global Menu
    .sharedstage.f2.canvas create rect $x $y [expr $x+70] [expr $y+20] \
       -fill $pcol -tag [list Participant $pnum user$pnum]
    .sharedstage.f2.canvas create text [expr $x+5] [expr $y+18] \
       -text $pnum -anchor sw -tag [list Participant $pnum] \
       -font $Menu(font) -fill White
}

# Conference Event Handler
proc MConferenceEventHandlerOn {} {
       global tpos Color
         gk_bind newUserArrived {
                 set newuser %U
                 set num [users remote.$newuser.usernum]
              set col [users remote.$newuser.color]
```

```
            set tpos [expr $tpos + 180 ]
            MDrawParticipant $num $col $tpos 20
            vtkActorCollection allActors(%U)
            vtkActorCollection copiedActors(%U)
            vtkActorCollection activeActors(%U)
            vtkActorCollection lastActors(%U)
            vtkPicker picker(%U)
            MAssignColorToProperty prop(%U) $Color($col)
        }
        gk_bind userDeleted {
                set olduser %U
            MResetParticipants
            allActors(%U) RemoveAllItems; allActors(%U) Delete
            copiedActors(%U) RemoveAllItems; copiedActors(%U) Delete
            activeActors(%U) RemoveAllItems; activeActors(%U) Delete
            lastActors(%U) RemoveAllItems; lastActors(%U) Delete
            picker(%U) Delete
            prop(%U) Delete
        }
}

proc MUpdateStructureTree {} {
    global myid

    MDeleteAllParticipants
    set tpos 20
    MDrawParticipant [users get local.usernum] [users get local.color]\
        $tpos 20
    foreach i [users keys remote] {
      set tpos [expr $tpos + 180 ]
       MDrawParticipant [users get remote.$i.usernum] \
            [users get remote.$i.color] $tpos 20
    }

    .sharedstage.f2.canvas delete -tag line
    .sharedstage.f2.canvas delete -tag leaf
    .sharedstage.f2.canvas delete -tag link
    set xposition 1; set yposition 1.5
    MUpdateStructure $myid [users get local.color] $xposition $yposition
    set xposition [expr $xposition + 6 ]
    foreach i [users keys remote] {
      MUpdateStructure $i [users get remote.$i.color] $xposition $yposition
      set position [expr $xposition + 6 ]
    }

    foreach i [GkEnv_ActorLink keys] {
      set coord [.sharedstage.f2.canvas coords $i]
       set x [lindex $coord 0]
      set y [expr [lindex $coord 1]+12]
      set x1 [expr $x + 10]
      foreach j [GkEnv_ActorLink get $i] {
          set y1 [expr $y + 3]
          if {$j == $myid } { set color [users get local.color]
          } else { set color [users get remote.$j.color] }
          catch {.sharedstage.f2.canvas create rect $x $y $x1 $y1 \
             -fill $color -outline $color -tag [list link ${i}_${j}] }
          set y [expr $y+5]
      }
    }
}

proc MUpdateStructure {id color xposition yposition} {
    global allActors lineCount

    set unit 30
```

```
    set origin [MCountLinesOfStructure allActors($id) $id]
    set x 0
    allActors($id) InitTraversal
    set part [allActors($id) GetNextItem]
    while { $part != "" } {
        set x [expr [MPrintStructure $part [expr $xposition+1] $x $origin
$color 0] + $x]
       set part [allActors($id) GetNextItem]
    }
    if { $yposition < $origin } {
       .sharedstage.f2.canvas create line [expr $xposition*$unit] \
                                   [expr $yposition*$unit] \
                                   [expr $xposition*$unit] \
                                   [expr $origin * $unit ] -tag line
    }
}

proc MPrintStructure {actor depth dist orig color level} {
    regexp {[^,]+,[^,]+,([^\)]*)\)} $actor match _n
    set unit 30
    set x 0
    if { [$actor GetClassName] != "vtkAssembly" } {
        .sharedstage.f2.canvas create line \
                             [expr [expr $depth-1]*$unit] \
                                  [expr [expr $orig*$unit]-5] \
                                  [expr [expr $depth-1]*$unit] \
                                  [expr [expr $orig - $dist] * $unit ] \
                                  [expr $depth * $unit ] \
                                  [expr [expr $orig - $dist] * $unit ] -tag line
        .sharedstage.f2.canvas create rect \
                                   [expr [expr $depth * $unit]-5 ] \
                             [expr [expr [expr $orig - $dist] * $unit ]-5 ] \
                                  [expr [expr $depth * $unit]+5 ] \
                                [expr [expr [expr $orig - $dist] * $unit ]+5 ] \
                                  -fill $color -tag [list leaf $actor]
        if {$level == 0} {
           .sharedstage.f2.canvas bind $actor <1> \
              "MStartRubberLineOnCanvas %W %x %y $actor"
           .sharedstage.f2.canvas bind $actor <B1-Motion> \
              "MDrawRubberLineOnCanvas %W %x %y"
           .sharedstage.f2.canvas bind $actor <ButtonRelease-1> \
              "MChangeOwner %W %x %y $actor"

           .sharedstage.f2.canvas bind $actor <2> \
              "MStartRubberLineOnCanvas %W %x %y $actor"
           .sharedstage.f2.canvas bind $actor <B2-Motion> \
              "MDrawRubberLineOnCanvas %W %x %y"
           .sharedstage.f2.canvas bind $actor <ButtonRelease-2> \
              "MStartLinkActor %W %x %y $actor $color"

           .sharedstage.f2.canvas bind $actor <3> "MUnLinkActor $actor"
       } else {
           .sharedstage.f2.canvas bind $actor <1> ""
           .sharedstage.f2.canvas bind $actor <B1-Motion> ""
           .sharedstage.f2.canvas bind $actor <ButtonRelease-1> ""

           .sharedstage.f2.canvas bind $actor <2> ""
           .sharedstage.f2.canvas bind $actor <B2-Motion> ""
           .sharedstage.f2.canvas bind $actor <ButtonRelease-2> ""

           .sharedstage.f2.canvas bind $actor <3> ""
       }

        .sharedstage.f2.canvas create text \
```

```
                                [expr [expr $depth * $unit]+15 ] \
                        [expr [expr [expr $orig - $dist] * $unit ]+1 ] \
                        -text $_n -fill $color -tag leaf -justify left \
            -font -*-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1

        incr x;
    } else {
        .sharedstage.f2.canvas create line \
                        [expr [expr $depth-1]*$unit] \
                            [expr [expr $orig*$unit]-5] \
                            [expr [expr $depth-1]*$unit] \
                            [expr [expr $orig - $dist] * $unit ] \
                            [expr $depth * $unit ] \
                            [expr [expr $orig - $dist] * $unit ] -tag line
        .sharedstage.f2.canvas create rect \
                            [expr [expr $depth * $unit]-5 ] \
                          [expr [expr [expr $orig - $dist] * $unit ]-5 ] \
                            [expr [expr $depth * $unit]+5 ] \
                          [expr [expr [expr $orig - $dist] * $unit ]+5 ] \
                            -fill White -tag [list leaf $actor]
        if {$level == 0} {
            .sharedstage.f2.canvas bind $actor <1> \
              "MStartRubberLineOnCanvas %W %x %y $actor"
            .sharedstage.f2.canvas bind $actor <B1-Motion> \
              "MDrawRubberLineOnCanvas %W %x %y"
            .sharedstage.f2.canvas bind $actor <ButtonRelease-1> \
              "MChangeOwner %W %x %y $actor"

            .sharedstage.f2.canvas bind $actor <2> \
              "MStartRubberLineOnCanvas %W %x %y $actor"
            .sharedstage.f2.canvas bind $actor <B2-Motion> \
              "MDrawRubberLineOnCanvas %W %x %y"
            .sharedstage.f2.canvas bind $actor <ButtonRelease-2> \
              "MStartLinkActor %W %x %y $actor $color"

            .sharedstage.f2.canvas bind $actor <3> "MUnLinkActor $actor"
        } else {
            .sharedstage.f2.canvas bind $actor <1> ""
            .sharedstage.f2.canvas bind $actor <B1-Motion> ""
            .sharedstage.f2.canvas bind $actor <ButtonRelease-1> ""

            .sharedstage.f2.canvas bind $actor <2> ""
            .sharedstage.f2.canvas bind $actor <B2-Motion> ""
            .sharedstage.f2.canvas bind $actor <ButtonRelease-2> ""

            .sharedstage.f2.canvas bind $actor <3> ""
        }

         .sharedstage.f2.canvas create text \
                                [expr [expr $depth * $unit]+15 ] \
                        [expr [expr [expr $orig - $dist] * $unit ]+1 ] \
                        -text $_n -fill $color -tag leaf -justify left \
            -font -*-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1

        [$actor GetParts] InitTraversal
        incr x
        set part [[$actor GetParts] GetNextItem]
        while { $part != "" } {
                set x [expr \
[MPrintStructure $part [expr $depth+1] $x [expr $orig-$dist] $color 1] + $x]
                set part [[$actor GetParts] GetNextItem]
        }
    }
    return $x
}
```

```
proc MCountLinesOfStructure {actor id} {
    global allActors

    set x 0
    if { [$actor GetClassName] != "vtkAssembly" && \
              [$actor GetClassName] != "vtkActorCollection" } {
       incr x;
    } else {
       if { $actor == "allActors($id)" } {
           $actor InitTraversal
           incr x;
           set part [allActors($id) GetNextItem]
           while { $part != "" } {
              set x [expr [MCountLinesOfStructure $part $id] + $x]
               set part [allActors($id) GetNextItem]
           }
       } else {
           [$actor GetParts] InitTraversal
           incr x;
           set part [[$actor GetParts] GetNextItem]
           while { $part != "" } {
              set x [expr [MCountLinesOfStructure $part $id] + $x]
              set part [[$actor GetParts] GetNextItem]
           }
       }
    }
    return $x
}

# Adding actor to all render windows except shared stage
proc MAddActorToAlmostAllRenderWindow {actor usrid} {
    global ren renWin
    foreach index [array names ren *,$usrid] {
       if {[[$ren($index) GetActors] IsItemPresent $actor] == 0} {
             $ren($index) AddActor $actor
       }
    }
    foreach index [array names renWin *,$usrid] {
        $renWin($index) Render
    }
}

# Removing actor from all render windows except shared stage
proc MRemoveActorFromAlmostAllRenderWindow {actor usrid} {
    global ren renWin
    foreach index [array names ren *,$usrid] {
       if { $ren($index) != "ren_sharedstage" } {
             $ren($index) RemoveActor $actor
       }
    }
    foreach index [array names renWin *,$usrid] {
        $renWin($index) Render
    }
}


# Changing ownership
proc MChangeActorsOwnership {actor usrid} {
    global ren renWin

    MUnLinkActor $actor
    activeActors([$actor GetId]) RemoveItem $actor
    allActors([$actor GetId]) RemoveItem $actor
    MRemoveActorFromAllRenderWindow $actor [$actor GetId]
```

```
    MRecursiveOwnerInfoChange $actor $usrid
    allActors($usrid) AddItem $actor
    MAddActorToAllRenderWindow $actor $usrid
}

proc MRecursiveOwnerInfoChange {actor usrid} {
    if { [$actor GetClassName] != "vtkAssembly" } {
       $actor SetId $usrid
       $actor SetProperty prop($usrid)
    } else {
       [$actor GetParts] InitTraversal
       set part [[$actor GetParts] GetNextItem]
       while {$part != "" } {
           MRecursiveOwnerInfoChange $part $usrid
           set part [[$actor GetParts] GetNextItem]
       }
       $actor SetId $usrid
       $actor SetProperty prop($usrid)
    }
}


# an interface routine to change actors ownership and link status
proc MStartRubberLineOnCanvas {w x y actor} {
    global canvas_lastx canvas_lasty

    gk_toAll .sharedstage.f2.canvas itemconfig $actor -width 2 \
       -outline red
    set temp_prop [$actor GetProperty]
    MChangeActorRepresentation $actor propActiveOnShared
    MUpdateMySharedRenderWindow
    catch {unset canvas_lastx}
    catch {unset canvas_lasty}
    set canvas_lastx [$w canvasx $x]
    set canvas_lasty [$w canvasy $y]

    MChangeActorRepresentation $actor $temp_prop
}

proc MDrawRubberLineOnCanvas {w x y} {
    global canvas_lastx canvas_lasty

    catch {$w delete -tag rubber_line}
    if { $x < 0 } {$w xview scroll -1 units}
    if { $x > [$w cget -width] } {$w xview scroll  1 units}
    if { $y < 0 } {$w yview scroll -1 units}
    if { $y > [$w cget -height] } {$w yview scroll  1 units}
    set x [$w canvasx $x]
    set y [$w canvasy $y]
    $w create line $canvas_lastx $canvas_lasty $x $y \
       -tag rubber_line -arrow last
}

proc MChangeOwner {w x y actor} {
    global canvas_lastx canvas_lasty

    gk_toAll .sharedstage.f2.canvas itemconfig $actor -width 1 -outline black
    MUpdateMySharedRenderWindow
    catch {$w delete -tag rubber_line}
    set x [$w canvasx $x]
    set y [$w canvasy $y]
    set usr_rect [lindex [$w find overlapping $x $y $x $y] 0]
    set tags [$w gettags $usr_rect]
    if { [lindex $tags 0] == "Participant" } {
       set usrid [lindex $tags 1]
```

```
        gk_toAll MChangeActorsOwnership $actor $usrid
        gk_toAll MUpdateStructureTree
    }
}

proc MStartLinkActor {w x y actor color} {
    global canvas_lastx canvas_lasty

    gk_toAll .sharedstage.f2.canvas itemconfig $actor -width 1 -outline black
    MUpdateMySharedRenderWindow
    catch {$w delete -tag rubber_line}
    set x [$w canvasx $x]
    set y [$w canvasy $y]
    if { [users get local.color] == $color } {return}
    set usr_rect [lindex [$w find overlapping $x $y $x $y] 0]
    set tags [$w gettags $usr_rect]
    if { [lindex $tags 0] == "Participant" } {
        set usrid [lindex $tags 1]
        MLinkActor $actor $usrid $color
    }
}

proc MLinkActor {actor usrid color} {

    set env_value [GkEnv_ActorLink get $actor]
    if { [lsearch -exact $env_value $usrid] < 0 } {
        GkEnv_ActorLink set $actor [lappend env_value $usrid]
         MAddActorToAlmostAllRenderWindow $actor $usrid
        gk_toAll MUpdateStructureTree
    }
}

proc MUnLinkActor {actor} {
    global myid

    set env_value [GkEnv_ActorLink get $actor]
    if { [lsearch -exact $env_value $myid] >= 0 } {
        GkEnv_ActorLink set $actor [ldelete [GkEnv_ActorLink get $actor] $myid]
        if { [GkEnv_ActorLink get $actor] == "" } {GkEnv_ActorLink delete
$actor}
         MRemoveActorFromAlmostAllRenderWindow $actor $myid
        gk_toAll MUpdateStructureTree
    }
}

proc ldelete { list value } {
    set ix [lsearch -exact $list $value]
    if {$ix >= 0} {
        return [lreplace $list $ix $ix]
    } else {
        return $list
    }
}

gk_newenv -share -bind GkEnv_ActorLink
```

C.2.8 gkenv.tcl

```
# This file contains scripts that manages synchronisation of the Shared Stage
# View using the GroupKit's environment feature.
gk_newenv -share -bind GkEnv_flag
GkEnv_flag set cflag 0

GkEnv_flag bind changeEnvInfo {
```

```
    if {"%K" == "cflag"} {
    set fop [GkEnv_flag get s.fop]
    set pos [GkEnv_flag get s.pos]
    set vup [GkEnv_flag get s.vup]
    set vpn [GkEnv_flag get s.vpn]
    set cen [GkEnv_flag get s.cen]
    set cpr [GkEnv_flag get s.cpr]
        set scam [$ren(shared,$myid) GetActiveCamera]
        $scam SetFocalPoint [lindex $fop 0] [lindex $fop 1] [lindex $fop 2]
        $scam SetPosition [lindex $pos 0] [lindex $pos 1] [lindex $pos 2]
        $scam SetViewUp [lindex $vup 0] [lindex $vup 1] [lindex $vup 2]
        $scam SetViewPlaneNormal [lindex $vpn 0] [lindex $vpn 1] [lindex $vpn
2]
        $scam SetWindowCenter [lindex $cen 0] [lindex $cen 1]
        $scam SetClippingRange [lindex $cpr 0] [lindex $cpr 1]
        $renWin(shared,$myid) Render
    }
}

proc MUpdateSharedCamera {} {
    global ren renWin myid
    set cam [$ren(shared,$myid) GetActiveCamera]
    GkEnv_flag set s.fop [$cam GetFocalPoint]
    GkEnv_flag set s.pos [$cam GetPosition]
    GkEnv_flag set s.vup [$cam GetViewUp]
    GkEnv_flag set s.vpn [$cam GetViewPlaneNormal]
    GkEnv_flag set s.cen [$cam GetWindowCenter]
    GkEnv_flag set s.cpr [$cam GetClippingRange]
}
```